313-CD-006-003

EOSDIS Core System Project

Version 2.0 CSMS/SDPS Internal Interface Control Document for the ECS Project

This document has not yet been approved by the Government for general use or distribution.

DRAFT UPDATE

February 1998

Raytheon Systems Company
Upper Marlboro, Maryland

This page intentionally left blank.

Version 2.0 CSMS/SDPS Internal Interface Control Document for the ECS Project

DRAFT UPDATE

February 1998

Prepared Under Contract NAS5-60000
CDRL Item #051

RESPONSIBLE ENGINEER

Eric J. Justice /s/ 2/27/98

Eric J. Justice Date
EOSDIS Core System Project

SUBMITTED BY

Mary S. Armstrong /s/ 2/27/98

Mary Armstrong, Deputy Director of Dev. Dept. Date
EOSDIS Core System Project

Raytheon Systems Company
Upper Marlboro, Maryland

This page intentionally left blank.

Preface

This document is a formal contract deliverable with an approval code 1. It requires Government review and approval prior to acceptance and use. This document is under ECS contractor configuration control. Once this document is approved, Contractor approved changes are handled in accordance with Class I and Class II change control requirements described in the EOS Configuration Management Plan, and changes to this document shall be made by document change notice (DCN) or by complete revision.

Any questions or proposed changes should be addressed to:

Data Management Office
The ECS Project Office
Raytheon Systems Company
1616 McCormick Drive
Upper Marlboro, MD 20774-5372

This page intentionally left blank.

Abstract

This document provides a set of interface scenarios that describe how the Version 2.0 SDPS/CSMS interact to execute end-to-end system threads. For each scenario a domain (or end user) view and a component interaction view is presented. This document is intended to be used by application developers, system developers and system maintenance engineers to understand how SDPS/CSMS components interact to perform key system functions.

Keywords: external interface, internal interface, public class, private class, class category, key mechanism, system-level scenario, scenario primitive, interface class, distributed object

This page intentionally left blank.

Change Information Page

List of Effective Pages	
Page Number	**Issue**
Title	Submitted As Draft Update
iii through x	Submitted As Draft Update
1-1 and 1-2	Submitted As Draft Update
2-1 through 2-2	Submitted As Draft Update
3-1 and 3-130	Submitted As Draft Update
AB-1 through AB-18	Submitted As Draft Update

Document History			
Document Number	**Status/Issue**	**Publication Date**	**CCR Number**
313-CD-006-001	Preliminary	October 1996	95-0778
313-CD-006-002	Submitted As Final	March 1996	96-0217
313-CD-006-003	Submitted As Draft	January 1998	97-1765
313-CD-006-003	Submitted As Draft Update	February 1998	98-0234

This page intentionally left blank.

Contents

List of Figures

List of Tables

Abbreviations and Acronyms

1. Introduction

1.1 Identification

This Version 2.0 CSMS/SDPS Internal Interface Control Document (ICD) for the ECS Project, Contract Data Requirement List (CDRL) item 051, with requirements specified in Data Item Description (DID) 313/DV3, is a required deliverable under the Earth Observing System (EOS) Data and Information System (EOSDIS) Core System (ECS), Contract (NAS5-60000).

1.2 Scope

The Version 2.0 Internal ICD specifies software interfaces internal to the CSMS/SDPS software architecture. It defines Version 2.0 services in the context of system level scenarios. The relationships and interactions between the Version 2.0 CSCIs are presented. This document also describes how ECS infrastructure services are used by the ECS internal applications.

This document addresses all interface classes from SDPS and CSMS CSCIs. External interfaces are mapped to the internal ECS object(s) that provide the service.

This document describes the ECS system in terms of its support of three primary scenarios. These scenarios were based on the normal support of three primary EOS instruments:

- MODIS
- Landsat 7
- ASTER

1.3 Document Organization

The document is organized to describe the Version 2.0 internal interfaces.

Section 1 provides information regarding the identification, scope, status, and organization of this document.

Section 2 provides a listing of the related documents which were used as source information for this document.

Section 3 contains the system level scenarios that illustrate the interactions between the ECS CSCIs. This section also provides an overview of the interface modeling approach to document the internal interfaces.

1.4 Status and Schedule

This is a preliminary draft of the Version 2.0 CSMS/SDPS Internal Interface Control Document for the ECS Project. It meets the milestone specified in the Contract Data Requirements List (CDRL) for Critical Design Review (CDR) of NASA contract NAS5-60000.

This page intentionally left blank.

2. Related Documentation

2.1 Parent Documents

TBD.

2.2 Applicable Documents

TBD.

2.3 Information Documents Not Referenced

TBD.

This page intentionally left blank.

3. Interface Scenarios

3.1 Overview

The purpose of this section is to document how ECS works to fulfill its mission. The ECS mission is, in its essence, to manage Earth Science-related data in the following ways:

- to receive data from external sources,
- to save that data in either long-term or permanent storage,
- to produce higher level data products from the received data, and
- to support access to the data by Earth Scientists as well as other interested clients.

ECS is a complex software system that comprises nine subsystems. Each of those subsystems comprises a set of software programs (COTS and custom built) that work together to exchange information and control the management of Earth Science-related data.

One of the best ways to document how a complex system such as ECS works is to follow a specific thread of functionality, or scenario, tracing how the ECS Clients (both human and software) and internal ECS ***components*** interact in support of that scenario. The interaction between the ECS components can be understood by focusing on how the interfaces offered by the ECS components are used in support of the system functionality required to support the given scenario.

As stated above, this section is intended to document how the ECS system works together to fulfill its mission. This section should be viewed as documenting one facet of a multi-faceted problem. In order to get a more complete view of precisely how each ECS component performs its role, the reader should also reference the design material presented by each of the ECS components. This material can be found in DID-305[1]. Table 3.1-1 maps the subsystems and their components to their appropriate interface process. Indeed, this document and DID 305 should be used in conjunction with each other. DID 305 provides a statement of what the components are providing and how they provide it. This section will document how those components work together to provide a complete system.

[1] DID-305 refers to ECS Document: 305-CD-100-001, Version 2.0 Segment/Design Specification for the ECS Project.

Table 3.1-1. ECS Subsystem and Component Design Overviews (1 of 2)

Subsystem (CI)	Component	Interface Process
CLS	B0SOT	xims
	DAAC Desktop	EcClDtDesktopDaacUser
	Science Desktop	EcClDtDesktopSciUser
	User Profile Gateway	EcClDtUserProfileGateway
	DARTool	DART
	EOSView	EOSView
CSS	Subscription Server	EcSbSubServer
	Subscription Server GUI	EcSbGui
	Ftp Server	ftp_popen
	L7 Gateway	EcCsLandsat7Gateway
	ASTER DAR Comm Gateway	EcGwDARServer
	Science Data Server Subsetting	EcDsSdHdfeosServer
	Sybase (COTS for SBSRV)	
DMS	V0 Gateway	EcDmV0ToEcsGateway
	Data Dictionary	EcDmDictServer
	Sybase (COTS for DDICT)	
DPS	Autosys (COTS)	event_demon
	Job Management	EcDpPrJobMgmt
	DPR Execution	EcDpPrEM, EcDPPrDM, EcDpPrRunPge
	DPREP	EcDpPrAm1AncillaryDPREP, EcDpPrReplaceEphemeris
	QA Monitor	EcDpPrQaMonitorGUI
	AITTL	EcDpAtMgr
DSS (SDSRV)	Science Data Server	EcDsScienceDataServer
	Science Data Server Operator GUI	EcDsSdsrvGui
	Sybase/SQS (COTS for SDSRV)	
DSS (DDSRV)	Document Data Server (Not Part of 2.0)	
DSS (DDIST)	Data Distribution Server	EcDsDdRequestMgrMain
	Data Distribution Operator GUI	EcDsDdistGui
	Sybase (COTS for DDIST)	
DSS (STMGT)	Archive	EcDsStArchiveServer
	Print Server	EcDsStPrintServer
	StagingDisk	EcStStagingDiskServer
	Staging Monitor	EcDsStStagingMonitorServer
	Storage Management Operator GUI	EcDsStmgtGui
	Ftp Server (for Ingest)	EcDsStIngestFtpServer
	Ftp Server (for DDIST)	EcDsStFtpDisServer
	8mm Tape	EcDsSt8mmServer
	D3 Tape	EcDsStD3TapeServer
	Pull Monitor	EcDsStPullMonitorServer

Table 3.1-1. ECS Subsystem and Component Design Overviews (2 of 2)

Subsystem (CI)	Component	Interface Process
INS	Auto Ingest	EcInAuto
	Polling Ingest	EcInPolling
	Media Ingest GUI	EcInGUI
	Interactive Ingest GUI	EcInInter
	Request Manager	EcInReqMgr
	Granule Server	EcInGran
	Sybase (COTS for configuration and state)	
IOS	Advertising Server	EcIoAdServer
	Sybase (COTS for ADSRV)	
MSS	User Registration Server	MsAcManager
	User Registration Server GUI	MsAcRegUserGUI
	Order Tracking Server	MsAcOrderManager
	Order Tracking GUI	MsAcOrderGUI
	Use Profile Database (Sybase COTS)	
	HPOV (COTS)	
PLS	Subscription Editor	EcPlSubsEdit
	Production Request Editor	EcPlPREditor_IF
	Subscription Manager	EcPlSubMgr
	Production Planning Workbench	EcPlWb
	Resource Planning Workbench Editor	EcPlRpRe
	Resource Planning Workbench Reservation Editor	EcPlRpSi
	Resource Planning Workbench Timeline	EcPlRpTl
	Workbench Timeline Tool	EcPlTL
	Sybase (COTS for PDPS database)	

3.2 Scenario Approach

Sections 3.3 - 3.5 document the ECS system, in terms of its support of three primary scenarios. These scenarios are based on the normal support of three primary EOS instruments:

- MODIS,
- Landsat 7, and
- ASTER.

We will show how ECS supports each of these missions from two perspectives: The domain view and the component view. The domain view will break the scenario into a sequence of activities that are based on what happens from the Operational or Science Data perspective. This view presents how ECS-external users and systems interact with ECS as well as looking at how the science data is managed within ECS. This view does not present the details of specific process interactions, or interface definition and utilization.

Then we will present a more detailed set of interactions that describe each interface usage between ECS components. Each interaction between components will be documented, in terms of how and why.

Each of the scenarios documented here has been partitioned into primary threads of activity. Each thread, or chunk, of the scenario is documented independently in order to simplify the scenarios.

3.2.1 Scenario Presentation Approach

This section describes how this paper will present the ECS support of each scenario. As mentioned above, each Scenario will be partitioned into a sequence of "chunks" of activity. Each of those chunks will be documented in the same manner. The following paragraphs define that documentation approach.

Scenario Description: First, each scenario will be described from the science mission perspective. The primary system functions that are being exercised are identified.

Scenario Preconditions: All activities that must have been performed prior to the execution of the scenario are identified.

Scenario Partitions: The scenario threads are identified and described.

Scenario Thread Interaction Diagram: A diagram is presented for each Scenario Thread. This diagram shows external system, ECS User, DAAC Operator and ECS-internal subsystem interactions. The notation of the diagram allows for the interactions to be labeled with numbers and short terms. The arrow numbering will use the convention of a letter, representing the Thread within the Scenario, and a sequence number (e.g. A.1, A.2, B.2, B.3). The mechanism of the interactions (e.g. Distributed Object invocation, Asynchronous interaction, RPC invocation and GUI interaction) will be identified by the interaction arrow presentation style.

Interaction Table - Domain View: Each Scenario Thread will be documented in a table which describes the interactions presented in the Scenario Thread Interaction Diagram. These interactions are not the detailed definitions of how the interactions are fulfilled, but rather that they need to occur. This table will further specify the interactions as each row will represent an interaction. The columns in the table delimit how each interaction will be defined. The Interaction Table - Domain View will include the following columns:

> Step: An identifier of the step within the Scenario Thread. Each step is identified by a "x.y' label, where x is a letter referring to the Thread within the scenario, and y is a sequence number.
>
> Interaction: The name of the interaction that is occurring.
>
> Interface Client: The Client of the interaction. This can be viewed as who is asking the question, or who is stimulating the action. Included in this column are Users, Operators, External Systems and ECS subsystems or components.

Interface Provider: All Interaction is described in terms of exercising well-defined interfaces. Those interfaces are offered by some entity in the system. The interface provider is how we are identifying the entity offering the interface. The Interface Provider is not only responsible for offering the interface, but for ensuring that the interface is met. They are doing the action required, perhaps collaborating with other system entities.

Data Issues: This column describes any special Data related issues. Here we are describing the data types, volumes and frequencies, as well as the current source of the data used in the system.

Preconditions: Any special preconditions that must have been met for the interaction to be successful are called out here.

Description: A textual description of what is occurring during this step. This will describe what generally occurs during the interaction, as well as its application in this scenario step.

Component Interaction Table: Each Scenario Thread will be further documented in the Component Interaction Table. This table will specify each ECS component-level interaction that is required to support the steps in the Scenario Thread.

Each of these interactions will be numbered in a way that is consistent with the Scenario Thread that it is supporting. Specifically, each Component Interaction step will be numbered using the Scenario Thread that it is supporting, followed by a sequence number within that Scenario Thread step. For example, if there are three interactions required to fulfill Scenario Thread step A.3, those three steps will be numbered A.3.1, A.3.2 and A.3.3. The next Component interaction, which would be in support of Scenario Thread Step A.4, will be numbered A.4.1. Please note that if no component interaction is required to fulfill a Scenario Thread Step (i.e. - only human-to-human interaction), there will be no component interaction steps. Therefore, in the Component Interaction steps, a Scenario Thread might be skipped. For example, it is reasonable to have a Component Interaction sequence skip Scenario Thread A.3, and thereby have the following sequence of Component Interaction steps: A.2.1, A.2.2, A.4.1.

Each row in the Component Interaction Table defines a step in how the system supports the capability. The columns in the Component Interaction Table are:

Step: An identifier of the step within the Scenario Thread. Each step is identified as described above.

Event: A label of what is occurring.

Interface Client: The Client of the interaction. This can be viewed as who is asking the question, or who is stimulating the action. Included in this column are Users, Operators, External Systems and ECS components. Where ECS components are the Interface Clients, the specific component process is identified.

Interface Provider: This identifies the entity in the system that is providing the interface used to perform some capability. Interface Providers are primarily ECS Components, which will be identified by the component process name.

Interface Mechanism: This column identifies how the interface is accomplished. It defines the low level (normally software) mechanism used by the Interface Client and Provider to exchange necessary information.

Description: This column contains text describing what is occurring during this step. It describes what is occurring in the context of this scenario thread. It describes not only what is happening, but how it is happening and how the client knows how to ask for it.

3.3 MODIS Scenario

3.3.1 Scenario Description

This scenario shows how the ECS supports standard MODIS processing. Generally, MODIS data is made available to ECS when MODIS files are placed into a predetermined directory on a predetermined host that ECS polls for periodically. ECS detects the availability of new Level 0 MODIS data via a PDR (Product Delivery Record) file. ECS then ingests and archives that Level 0 granule, and a sequence of standard production algorithms is run based on that new data.

This scenario modifies the standard MODIS support in several ways. First, normally MODIS L0 data is processed into MODIS L1A data. However, the algorithm to perform this processing takes a significant amount of time to run. Since this time invested in standard processing of MODIS L0 into MODIS L1A is not conducive to an effective demonstration, that part of standard production has been replaced with the insertion of a synthetic PGE (PGE01syn) into the Production Plan. This synthetic PGE processes previously inserted MOD01synth and MOD03synth data into MOD01 and MOD03 granules. These MOD01 granules, along with ancillary data, are automatically input to PGE02 to produce MOD02 data. The MOD02 and MOD03 data are input to the PGE08, which produces MOD029 granules.

The following diagram illustrates the relationships between the data types and PGEs used in the MODIS Scenario.

MODIS PGEs Demonstrated

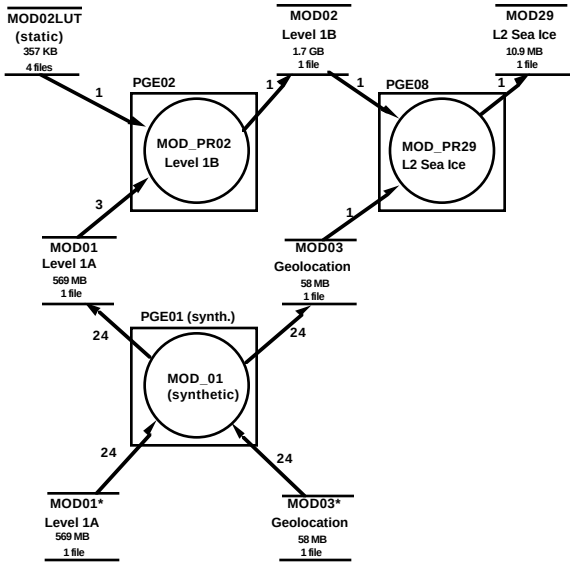

Figure 3.3.1-1. MODIS Scenario PGE/Data Relationship

The following system functionality is exercised in this scenario:

- Automatic Scheduling of PGE execution
- Archive Data as PGE Inputs
- Chaining of PGEs
- Concurrent PGE Execution
- Access to ECS produced data
- Standing Order Support, including distributing data to users electronically
- Support of failed PGEs.

3.3.2 Scenario Preconditions

The following ESDTs have been inserted into the ECS:

MOD01syn (copy of MOD01 data, used as input to PGE01synth)
MOD03syn (copy of MOD02 data, used as input to PGE01synth)
MOD01 (L1A Raw Counts)
MOD02 (L1B Radiances)
MOD02LUT (MODIS Ancillary Lookup Tables data)
MOD03 (MODIS Geolocation data)
MOD029 (L2 Sea Ice Cover)
PGEEXE (PGE Execution Granule)
FAILPGE (Failure PGE)
PH (Product History)

The following PGEs have successfully been through the SSI&T process:

PGE01synth
PGE02
PGE08

Ancillary data granules inserted into Data Server.

MOD01syn and MOD03syn granules inserted into Data Server.

Subscription for MODIS failed PGEs entered on behalf of Instrument Team.

A Resource Plan created for resources needed for MODIS production.

A Production Plan has been created using the Production Planning Workbench. This Production Plan includes Production Requests for the PGE01synth, PGE02 and PGE08, with appropriate sequencing dependencies between them. The DPR for PGE01synth job in the plan includes references to the appropriate MOD01syn and MOD03syn granules. The DPRs for PGE02 and PGE08 have submitted subscriptions for the Insert events for appropriate input data.

Production Planning Workbench is already running on the DAAC Desktop.

3.3.3 Scenario Partitions

The MODIS scenario has been partitioned into the following threads:

Standing Order Submittal (Thread A) - This thread simply shows how the DAAC User Services will submit a standing order for MOD029 granules to be distributed via ftpPush to a science user.

Standard Production (Thread B) - This thread will show how the sequence of PGEs (PGE01synth, PGE02, PGE08) will execute in a chained fashion, with the output of one PGE being used as the input of the next.

Failed PGE Handling (Thread C) - This thread will show how the artifacts from a failed PGE will be collected and sent to the Instrument Team.

Data Access (Thread D) - This thread will show how the generated data products are available for user access. Also in this thread, the Standing Order, submitted in Thread 1, will be fulfilled.

3.3.4 Standing Order Submittal Thread

3.3.4.1 Interaction Diagram - Domain View

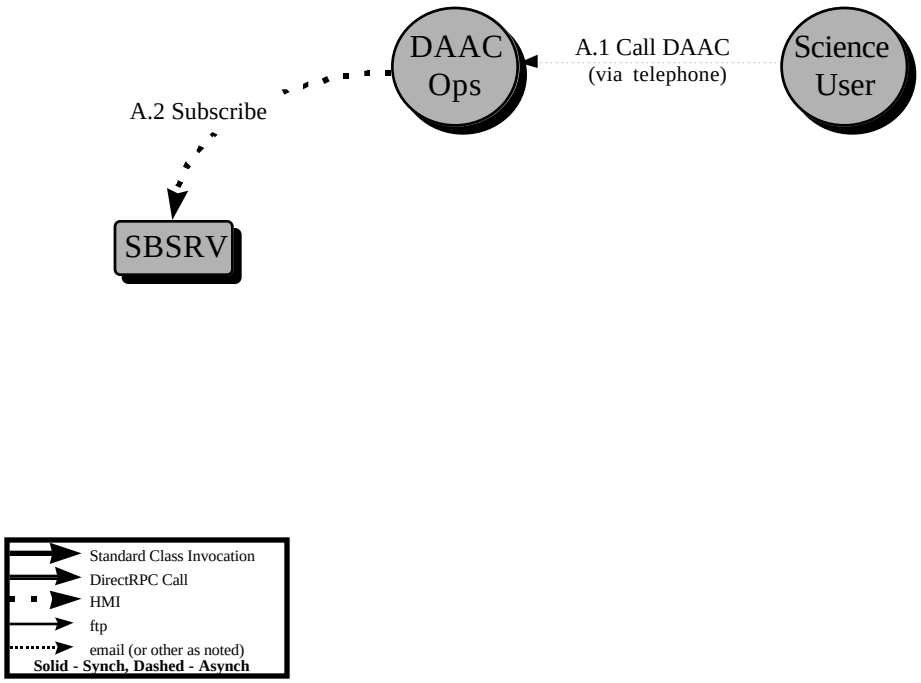

Figure 3.3.3.1-1. Standing Order Submittal Interaction Diagram

3.3.4.2 Interaction Table - Domain View

Table 3.3.4.2-1. Interaction Table - Domain View: MODIS scenario, Standing Order Submittal

Step	Interaction	Interface Client	Interface Provider	Data Issues	Preconditions	Description
A.1	Call DAAC User Services	Science User	DAAC		DAAC User Services Advertises Phone number. Email address available.	Science User calls DAAC User Services staff, requesting that all MOD029 granules are sent to his/her workstation, via ftpPush. Science User provides host name and address, directory for data placement, and username/password for ECS to use when placing data. Note: This could be performed via email as well as telephone.
A.2	Subscribe	DAAC User Services Representative	SBSRV		MOD029 ESDT, with Insert Event	DAAC User Services Representative submits Standing Order subscription for MOD029:Insert event. Action is to electronically push product to science user's machine.

3.3.4.3 Component Interaction Table

Table 3.3.4.3-1. Component Interaction Table: MODIS Scenario - Standing Order Submittal

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
A.2.1	Startup SBSRV GUI	DAAC User Services Representative	EcSbGui	Xterm	DAAC User Services Representative invokes SBSRV GUI application.
A.2.2	Create & Submit Subscription from GUI	DAAC User Services Representative	EcSbGui	Xterm	DAAC User Services Representative represents him/herself as the Science User. The DAAC Operator brings up the GUI and clicks button to create new subscription. A list of events is then displayed from which the op can choose to subscribe. DAAC Operator selects the MOD029:Insert Event for subscription. Only one action (besides notification), is available from the SBSRV at this time. FtpPush as a distribution mechanism is input via a GUI button. Other parameters required for FtpPush, including the Science User's host name, target directory, ftp user name, and ftp password, are input via the GUI.
A.2.3	Submit Subscription	EcSbGui	EcSbSubServer	Distributed Object	Submit the subscription to the Subscription Server. This is accomplished with the EcCISubscription interface class. The correct SBSRV is determined via a ServerUR, declared in configuration.
A.2.4	Persist a Subscription	EcSbSubServer	Sybase	CtLib	Subscription is stored in the Sybase Database.

3.3.5 Standard Production Thread

3.3.5.1 Interaction Diagram - Domain View

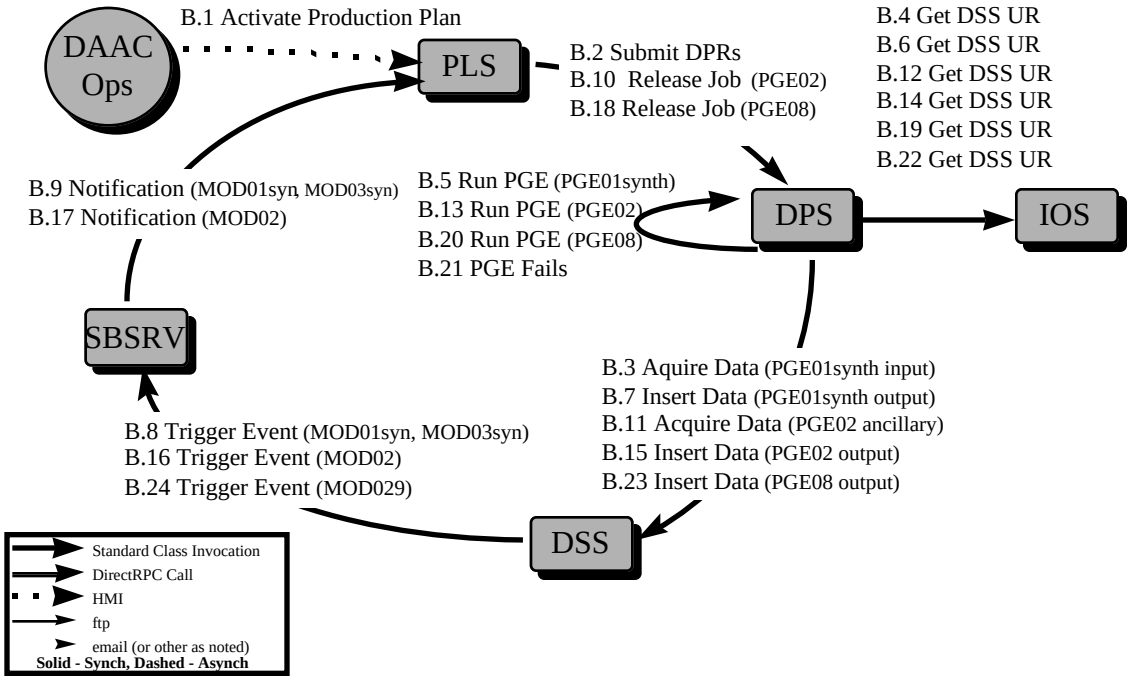

Figure 3.3.5.1-1. Standard Production Interaction Diagram

3.3.5.2 Interaction Table - Domain View

Table 3.3.5.2-1. Interaction Table - Domain View (MODIS scenario, Standard Production)

Step	Interaction	Interface Client	Interface Provider	Data Issues	Preconditions	Description
B.1	Activate Production Plan	DAAC Production Planner	PLS		PGEs passed SSI&T. Plan already created. Planner logged in to DAAC Desktop.	DAAC Production Planner activates a plan, which includes a DPR for PGE01synth with previously inserted granules. Plan created for PGE01synth, PGE02, PGE08, with input granules ID'd for PGE01syn, and subscriptions submitted for input data for PGE02 and PGE08.
B.2	Submit DPRs	PLS	DPS			DPRs for PGE01synth, PGE02 and PGE08 are submitted to DPS. All of these DPRs are submitted in an "Offhold" state, but PGE02 and PGE 08 have dependencies on the previous DPRs (PGE02 needs PGE01 DPR, PGE08 needs PGE01 and PGE02 DPRs).
B.3	Acquire Data	DPS	DSS	2 MOD01syn @569MB, 2 MOD01syn @187MB, and 4 MOD03syn @58MB, from MODIS IT		DPS submits Acquire Request for MOD01syn and MOD03syn, via ftpPush, for input to PGE01synth.
B.4	Get DSS UR	DPS	IOS			DPS gets the DSS UR from Advertiser.
B.5	Run PGE	DPS		2 MOD01syn @569MB, 2 MOD01syn @187MB, and 4 MOD03 @58MB produced		PGE01synth runs, creating MOD01 and MOD03 granules.
B.6	Get DSS UR	DPS	IOS			DPS gets the DSS UR from Advertiser.
B.7	Insert Data	DPS	DSS		MOD01 and MOD03 ESDTs.	Archive newly created MOD01 and MOD03 granules.

Table 3.3.5.2-1. Interaction Table - Domain View (MODIS scenario, Standard Production) (Continued)

Step	Interaction	Interface Client	Interface Provider	Data Issues	Preconditions	Description
B.8	Trigger Event	DSS	SBSRV			Trigger MOD01:Insert and MOD03:Insert events. Note that these are actually two different events, so there will be two independent events triggered, four times each (one for each granule).
B.9	Notification	SBSRV	PLS		PLS Subscriptions for MOD01:Insert and MOD03:Insert events	Send direct notification to PLS, notifying that there is a newly inserted MOD01 and MOD03 granules. Note that there are really 8 notifications received by PLS, one for each of the 4 MOD01 granules and one for each of the 4 MOD03 granules. Notifications includes the UR of the granules produced.
B.10	Release Job	PLS	DPS			PLS releases job containing PGE02.
B.11	Acquire Data	DPS	DSS	4 MOD02LUT @357KB, from MODIS IT	MOD02LUT ESDT. 4 granules previously inserted	DPS submits Acquire Request for the ancillary product, MOD02LUT, via ftpPush, for input to PGE02. Note that other input to PGE02 is the MOD01 granule that was created in step B.4.
B.12	Get DSS UR	DPS	IOS			DPS gets the DSS UR from Advertiser.
B.13	Run PGE			2 MOD02, 1.7GB & 200MB produced		PGE02 runs, creating MOD02 and MOD08 granules.
B.14	Get DSS UR	DPS	IOS			DPS gets the DSS UR from Advertiser.
B.15	Insert Data	DPS	DSS		MOD02 ESDT.	Archive newly created MOD02 and MOD08 granules.
B.16	Trigger Event	DSS	SBSRV			Trigger MOD02:Insert event.
B.17	Notification	SBSRV	PLS		PLS Subscription for MOD02:Insert event.	Send direct notification to PLS, notifying that there is a newly inserted MOD02 granule. Notification message includes the UR of the granule.
B.18	Release Job	PLS	DPS			PLS releases job containing PGE08.

Table 3.3.5.2-1. Interaction Table - Domain View (MODIS scenario, Standard Production) (Continued)

Step	Interaction	Interface Client	Interface Provider	Data Issues	Preconditions	Description
B.19	Get DSS UR	DPS	IOS			DPS gets the DSS UR from Advertiser.
B.20	Run PGE	DPS		1MOD029 @10.9 MB produced		PGE08 runs, creating MOD029 granule. All required inputs are still in DPS disk resources, so no Acquires are necessary.
B.21	PGE Fails	DPS				One instance of the PGE08 running fails, due to the need for night data, but all input data is during daylight. This is a planned PGE failure. Please pick up processing of this Failed PGE in Thread C of this MODIS scenario.
B.22	Get DSS UR	DPS	IOS			DPS gets the DSS UR from Advertiser.
B.23	Insert Data	DPS	DSS		MOD029 ESDT.	Archive newly created MOD029 granules.
B.24	Trigger Event	DSS	SBSRV			Trigger MOD029:Insert event. Completion of the support for Standing order will be shown in Thread D of this MODIS scenario.

3.3.5.3 Component Interaction Table

Table 3.3.5.3-1. Component Interaction Table: MODIS Scenario - Standard Production

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.1.1	Startup Planning Workbench	DAAC Operator - Planner	EcPlWb	GUI	DAAC Planner invokes the Planning workbench. While already running within the DAAC Desktop, the planner double clicks on the Planning Workbench icon.
B.1.2	Select Plan to Activate, Modify and Activate	DAAC Operator - Planner	EcPlWb	GUI	Planner interacts with Planning Workbench GUI to Select plan to activate (it was already created), modify it with DPRs for chaining PGE01synth, PGE02 and PGE08. Input granules for PGE01synth are identified in Production Plan. (Note scenario preconditions stated above.)

Table 3.3.5.3-1. Component Interaction Table: MODIS Scenario - Standard Production (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.1.3	Activate a Plan	EcPlWb	EcDpPrJobMgmt	rpc	Updated plan is activated, making it the current processing plan.
B.2.1	Submit DPRs	EcPlWb	event_demon	rpc (COTS)	DPRs (5 total: 1 containing PGE01synth, 2 containing PGE02 and 2 containing PGE08) in updated plan are submitted, to AutoSys, by DPS for dependent execution. PGE01synth job is automatically released, because all inputs are available and production rules have been met, because input granules were referenced in DPR.
B.2.2	Job Processing	event_demon	EcDpPrEM	command line	The job containing the PGE01synth begins processing.
B.2.3	Connect to SDSRV	EcDpPrEM	EcDsScienceDataServer	Distributed Object	Processing begins a session with the SDSRV by connecting, in order to acquire the PGE01synth. The correct SDSRV is determined by using the GranuleUR of the PGE granule, which is defined in the Production plan and is part of the DPR.
B.2.4	Add granules to Session	EcDpPrEM	EcDsScienceDataServer	Distributed Object	PRONG establishes the data context of the session with the SDSRV by adding PGE granule's UR. The GranuleUR of the PGE granule is added to the ESDTReferenceCollection.
B.2.5	Retrieve Granule Metadata from Inventory	EcDsScience DataServer	Sybase/SQS	CtLib	SDSRV completes establishing the data context by retrieving the metadata for the requested PGE granule from the Sybase/SQS database. The metadata for the PGE granule is passed back to the reference objects for each granule.

Table 3.3.5.3-1. Component Interaction Table: MODIS Scenario - Standard Production (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.2.6	Acquire Data	EcDpPrEM	EcDsScienceDataServer	Distributed Object	DPS requests granules by submitting an Acquire request for the PGE granule. The Acquire request is for an ftpPush of all granules in the ESDTReferenceCollection. This request is synchronous, meaning that the return of the submit call of the request will contain the results of the request. This means that the response will not be sent until the PGE granule files have been ftp'd to the DPS disks. This request asks for no distribution notice to be email'd. The Acquire request structure is hardcoded.
B.2.7	Create Staging Disk	EcDsScience DataServer	EcStStagingDiskServer	Distributed Object	SDSRV creates Staging Disk for the metadata file, which allocates space and passes back a reference to that disk space. The reference to the Staging Disk is determined from the ArchiveID metadata of the granule to be distributed. The amount of staging disk to request is a default, hardcoded as a SDSRV constant.
B.2.8	Create Metadata file	EcDsScience DataServer			The SDSRV creates a file containing the PGE granule's metadata before passing to Distribution.
B.2.9	Distribute Granules, Synchronous	EcDsScience DataServer	EcDsDdRequestMgrMain	Distributed Object	SDSRV submits a request to DataDistribution. The request includes, for each granule, a reference to the metadata file as well as all data files. Other parameters from the Acquire request are passed to DDIST.
B.2.10	Create Staging Disk	EcDsDdReque stMgrMain	EcStStagingDiskServer	Distributed Object	DDIST creates Staging Disk for the granule files in the archive. This allocates space and passes back a reference to that disk space. The reference to the Staging Disk is determined from the information passed by the SDSRV in the distribution request ,which was the ArchiveID metadata parameter of the granule to be staged. The amount of staging disk to request is calculated from the file sizes of archived files, in the information passed in the Distribution Request.

Table 3.3.5.3-1. Component Interaction Table: MODIS Scenario - Standard Production (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.2.11	STMGT Retrieve	EcDsDdRequestMgrMain	EcDsStArchiveServer	Distributed Object	DDIST requests that STMGT retrieve the PGE granule file that is archived. This will result in the file being staged to read-only cache disks. This means that all files needed to fulfill the distribution request are on disk, and ready to be copied. The correct archive object to request is determined from the information provided by the SDSRV in the distribution request. This returns references to the files in the read-only cache.
B.2.12	Link files to Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST links the files from the read-only cache into the staging disk.
B.2.13	Copy files to Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST copies the metadata files from the SDSRV's Staging Disk into the staging disk.
B.2.14	ftpPush Files	EcDsDdRequestMgrMain	EcDsStFtpDisServer	Distributed Object	DDIST now creates the Resource manager for ftp Pushes via a Resource Manager Factory. The correct resource manager is determined from the MediaType handed to the resource factory (ftpPush, in this case). The correct FtpServer is determined from configuration within the resource factory. The files, host, location, username and password are all determined from the information provided in the original Acquire request.
B.2.15	ftp Files	EcDsStFtpDisServer	EcDpPrEM	rpc	CSS performs the actual low level ftp of the PGE files.
B.3.1	Connect to SDSRV	EcDpPrDM	EcDsScienceDataServer	Distributed Object	PRONG begins a session with the SDSRV by connecting. The correct SDSRV is determined by using the GranuleUR of the input granule.
B.3.2	Add granules to Session	EcDpPrDM	EcDsScienceDataServer	Distributed Object	PRONG establishes the data context of the session with the SDSRV by adding the input granules (4 MOD01syn and 4 MOD03syn) to the session. The GranuleUR of the input granule is added to the ESDTReferenceCollection. Note that this sequence is performed for each input granule, one at a time.

Table 3.3.5.3-1. Component Interaction Table: MODIS Scenario - Standard Production (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.3.3	Retrieve Granule Metadata from Inventory	EcDsScience DataServer	Sybase/SQS	CtLib	SDSRV completes establishing the data context by retrieving the metadata for the requested granule from the Sybase/SQS database. The metadata for each granule is passed back to the reference objects for each granule.
B.3.4	Acquire Data	EcDpPrDM	EcDsScienceDataServer	Distributed Object	DPS requests granules by submitting an Acquire request for those granules. The Acquire request is for an ftpPush of all granules in the ESDTReferenceCollection. This request is synchronous, meaning that the return of the submit call of the request will contain the results of the request. This means that the response will not be sent until the granule files have been ftp'd to the DPS disks. This request asks for no distribution notice to be email'd. The Acquire request structure is hardcoded.
B.3.5	Create Staging Disk	EcDsScience DataServer	EcStStagingDiskServer	Distributed Object	SDSRV creates Staging Disk for metadata files, which allocates space and passes back a reference to that disk space. The reference to the Staging Disk is determined from the ArchiveID metadata of the granule to be distributed. The amount of staging disk to request is determined from collection level metadata from the ESDT's Descriptor file.
B.3.6	Create Metadata file	EcDsScience DataServer			For each granule referenced in the Acquire request, the SDSRV creates a file containing the granule's metadata before passing to Distribution.
B.3.7	Distribute Granules, Synchronous	EcDsScience DataServer	EcDsDdRequestMgrMain	Distributed Object	SDSRV submits a request to DataDistribution. The request includes, for each granule, a reference to the metadata file as well as all data files. Other parameters from the Acquire request are passed to DDIST.

Table 3.3.5.3-1. Component Interaction Table: MODIS Scenario - Standard Production (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.3.8	Create Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST creates Staging Disk for the granule files in the archive. This allocates space and passes back a reference to that disk space. The reference to the Staging Disk is determined from the information passed by the SDSRV in the distribution request , which was the ArchiveID metadata parameter of the granule to be staged. The amount of staging disk to request is calculated from the file sizes in the information passed in the Distribution Request.
B.3.9	STMGT Retrieve	EcDsDdRequestMgrMain	EcDsStArchiveServer	Distributed Object	DDIST requests that STMGT retrieve the granule file that is archived. This will result in the file being staged to read-only cache disks. This means that all files needed to fulfill the distribution request are on disk, and ready to be copied. The correct archive object to request is determined from the information provided by the SDSRV in the distribution request. This returns references to the files in the read-only cache.
B.3.10	Link files to Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST links the files from the read-only cache into the staging disk.
B.3.11	Copy files to Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST copies the metadata files from the SDSRV's Staging Disk into the staging disk.
B.3.12	ftpPush Files	EcDsDdRequestMgrMain	EcDsStFtpDisServer	Distributed Object	DDIST now creates the Resource manager for ftp Pushes via a Resource Manager Factory. The correct resource manager is determined from the MediaType handed to the resource factory (ftpPush, in this case). The correct FtpServer is determined from configuration within the resource factory. The files, host, location, username and password are all determined from the information provided in the original Acquire request.
B.3.13	ftp Files	EcDsStFtpDisServer	EcDpPrDM	rpc	CSS performs the actual low level ftp of the files.

Table 3.3.5.3-1. Component Interaction Table: MODIS Scenario - Standard Production (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.4.1	Get DSS UR	EcDpPrEM	EcIoAdServer	Distributed Object	If the DSS UR for this Metadata Configuration File (MCF) is not already known in the PDPS database, EM searchAdvertiser for a "GetQueriableParameters" service for the desired MCF data type. This is accomplished via the IoAdApprovedSearchCommand class. Since the Advertiser is based on a replicated database, no specification is required to select the proper Advertiser. The local one is used.
B.5.1	Get Metadata Configuration File	EcDpPrEM	EcDsScienceDataServer		DPS gets the metadata configuration file of the output data's ESDT (MOD01 and MOD03). Datatype and version are from PDPS database; correct client name is from configuration file.
B.5.2	Run PGE	EcDpPrRunPGE	PGE<PGE01syn>	command line	PGE01Synth is executed. Output files are placed in the output directory. The directory path is established by using a root, which was established by configuration, and the specific directory by the job id. This disk root is cross mounted by DPS, SDSRV and STMGT. This is to ensure that they are directly available to the DSS, for archival.
B.6.1	Get DSS UR	EcDpPrDM	EcIoAdServer	Distributed Object	If the DSS UR for this output data type is not already known in the PDPS database, DM searches the Advertiser for a "GetQueriableParameters" service for the desired output data type. This is accomplished via the IoAdApprovedSearchCommand class. Since the Advertiser is based on a replicated database, no specification is required to select the proper Advertiser. The local one is used.
B.7.1	Connect to SDSRV	EcDpPrDM	EcDsScienceDataServer	Distributed Object	PRONG begins a session with the SDSRV by connecting.

Table 3.3.5.3-1. Component Interaction Table: MODIS Scenario - Standard Production (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.7.2	Insert Data	EcDpPrDM	EcDsScienceDataServer	Distributed Object	PRONG requests that the newly created files for the MOD01 and MOD03 granules are inserted into the Data Server. An Insert request, containing the names of the files comprising the granule, is created for each granule. The structure of the Insert Request is hardcoded. SDSRV validates metadata and determines the archived names of the files. Note that these inserts occur one granule at a time.
B.7.3	STMGT Store	EcDsScience DataServer	EcDsStArchiveServer	Distributed Object	SDSRV requests that the files are archived. The archive server must be able to read the inserted files directly from the DPS disks that they are residing on. The correct archive object to request is determined from collection level metadata for the ESDT, defined in the ESDT's descriptor.
B.7.4	Adding a Granule to Inventory	EcDsScience DataServer	Sybase/SQS	CtLib	The validated metadata is parsed and added to the inventory of the SDSRV.
B.8.1	Trigger Event	EcDsScience DataServer	EcSbSubServer	Distributed Object	Upon successful insertion of MOD01 and MOD03 granules, the MOD01:Insert and the MOD03:Insert events are triggered, one per granule. The correct subscription server is determined from the SDSRV configuration. The correct events to trigger are determined from the events file, where they were stored when the ESDT was installed in the Data Server. Provided with the event triggering is the UR of the inserted granule.
B.8.2	Fire Subscriptions	EcSbSubServer	Sybase	CtLib	SBSRV queries the Sybase database determining which subscriptions need to be activated, or fired. Each query "hit" is an activated subscription and will execute independently.

Table 3.3.5.3-1. Component Interaction Table: MODIS Scenario - Standard Production (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.9.1	Asynchronous Direct Notification	EcSbSubServer	EcPlSubMgr		The SBSRV will notify PLS that there are new MOD01 and MOD03 granules available. The UR of the granule is passed in the notification to the user, along with a reference to the subscription that is being fulfilled. Direct Notification is to a Queuename (See Message Passing Key Mechansim) that PLS- Subscription Manager, provided when the subscription was submitted.
B.9.2	Connect to SDSRV	EcPlSubMgr	EcDsScienceDataServer	Distributed Object	Subscription Manager begins a session with the SDSRV by connecting, in order to determine the use of the new granule. The correct SDSRV is determined by using the GranuleUR in the notification message.
B.9.3	Add granules to Session	EcPlSubMgr	EcDsScienceDataServer	Distributed Object	Subscription Manager establishes the data context of the session with the SDSRV by adding new granule's UR. The GranuleUR of the PGE granule is added to the ESDTReferenceCollection.
B.9.4	Retrieve Granule Metadata from Inventory	EcDsScience DataServer	Sybase/SQS	CtLib	SDSRV completes establishing the data context by retrieving the metadata for the requested PGE granule from the Sybase/SQS database. The metadata for the PGE granule is passed back to the reference objects for each granule.
B.9.5	Inspect Granule Value Parameters	EcPlSubMgr	EcDsScienceDataServer	Distributed Object	Subscription Manager checks the new granule's metdata attributes (type, version, filesize and temporal range), to determine which, if any, jobs can use it as input.
B.10.1	Release Job	EcPlSubMgr	EcDpPrJobMgmt	rpc	Planning tells the Job Manager to release the job containing PGE02.
B.10.2	Force Start Job	EcDpPrJobMgmt	event_demon	rpc	Job containing PGE02 is released.
B.10.3	Job Processing	event_demon	EcDpPrEM	command line	The job containing the PGE02 begins processing.

Table 3.3.5.3-1. Component Interaction Table: MODIS Scenario - Standard Production (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.10.4	Connect to SDSRV	EcDpPrEM	EcDsScienceDataServer	Distributed Object	Processing begins a session with the SDSRV by connecting, in order to acquire the PGE02. The correct SDSRV is determined by using the GranuleUR of the PGE granule, which is defined in the Production plan and is part of the DPR.
B.10.5	Add granules to Session	EcDpPrEM	EcDsScienceDataServer	Distributed Object	PRONG establishes the data context of the session with the SDSRV by adding PGE granule's UR. The GranuleUR of the PGE granule is added to the ESDTReferenceCollection.
B.10.6	Retrieve Granule Metadata from Inventory	EcDsScience DataServer	Sybase/SQS	CtLib	SDSRV completes establishing the data context by retrieving the metadata for the requested PGE granule from the Sybase/SQS database. The metadata for the PGE granule is passed back to the reference objects for each granule.
B.10.7	Acquire Data	EcDpPrEM	EcDsScienceDataServer	Distributed Object	DPS requests granules by submitting an Acquire request for the PGE granule. The Acquire request is for an ftpPush of all granules in the ESDTReferenceCollection. This request is synchronous, meaning that the return of the submit call of the request will contain the results of the request. This means that the response will not be sent until the PGE granule files have been ftp'd to the DPS disks. This request asks for no distribution notice to be email'd. The Acquire request structure is hardcoded.
B.10.8	Create Staging Disk	EcDsScience DataServer	EcStStagingDiskServer	Distributed Object	SDSRV creates Staging Disk for the metadata file, which allocates space and passes back a reference to that disk space. The reference to the Staging Disk is determined from the ArchiveID metadata of the granule to be distributed. The amount of staging disk to request is a default, hardcoded as a SDSRV constant.
B.10.9	Create Metadata file	EcDsScience DataServer			The SDSRV creates a file containing the PGE granule's metadata before passing to Distribution.

Table 3.3.5.3-1. Component Interaction Table: MODIS Scenario - Standard Production (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.10.10	Distribute Granules, Synchronous	EcDsScience DataServer	EcDsDdRequestMgrMain	Distributed Object	SDSRV submits a request to DataDistribution. The request includes, for each granule, a reference to the metadata file as well as all data files. Other parameters from the Acquire request are passed to DDIST.
B.10.11	Create Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST creates Staging Disk for the granule files in the archive. This allocates space and passes back a reference to that disk space. The reference to the Staging Disk is determined from the information passed by the SDSRV in the distribution request ,which was the ArchiveID metadata parameter of the granule to be staged. The amount of staging disk to request is calculated from the file sizes of archived files, in the information passed in the Distribution Request.
B.10.12	STMGT Retrieve	EcDsDdRequestMgrMain	EcDsStArchiveServer	Distributed Object	DDIST requests that STMGT retrieve the PGE granule file that is archived. This will result in the file being staged to read-only cache disks. This means that all files needed to fulfill the distribution request are on disk, and ready to be copied. The correct archive object to request is determined from the information provided by the SDSRV in the distribution request. This returns references to the files in the read-only cache.
B.10.13	Link files to Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST links the files from the read-only cache into the staging disk.
B.10.14	Copy files to Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST copies the metadata files from the SDSRV's Staging Disk into the staging disk.

Table 3.3.5.3-1. Component Interaction Table: MODIS Scenario - Standard Production (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.10.15	ftpPush Files	EcDsDdRequestMgrMain	EcDsStFtpDisServer	Distributed Object	DDIST now creates the Resource manager for ftp Pushes via a Resource Manager Factory. The correct resource manager is determined from the MediaType handed to the resource factory (ftpPush, in this case). The correct FtpServer is determined from configuration within the resource factory. The files, host, location, username and password are all determined from the information provided in the original Acquire request.
B.10.16	ftp Files	EcDsStFtpDisServer	EcDpPrEM	rpc	CSS performs the actual low level ftp of the PGE files.
B.11.1	Connect to SDSRV	EcDpPrDM	EcDsScienceDataServer	Distributed Object	DPS begins a session with the SDSRV by connecting. The correct SDSRV is determined by using the GranuleUR of the granule from the SBSRV Notification.
B.11.2	Add granules to Session	EcDpPrDM	EcDsScienceDataServer	Distributed Object	PRONG establishes the data context of the session with the SDSRV by adding the input granules (4 MOD02LUT) to the session. The GranuleUR of each input granule is added to the ESDTReferenceCollection. Note that this sequence is performed for each input granule, one at a time.
B.11.3	Retrieve Granule Metadata from Inventory	EcDsScienceDataServer	Sybase/SQS	CtLib	SDSRV completes establishing the data context by retrieving the metadata for the requested granules from the Sybase/SQS database. The metadata for each granule is passed back to the reference objects for each granule.

Table 3.3.5.3-1. Component Interaction Table: MODIS Scenario - Standard Production (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.11.4	Acquire Data	EcDpPrDM	EcDsScienceDataServer	Distributed Object	DPS requests MOD02LUT ancillary granules by submitting an Acquire request for those granules. The Acquire request is for an ftpPush of all granules in the ESDTReferenceCollection. This request is synchronous, meaning that the return of the submit call of the request will contain the results of the request. This means that the response will not be sent until the granule files have been ftp'd to the DPS disks. This request asks for no distribution notice to be email'd. The Acquire request structure is hardcoded.
B.11.5	Create Staging Disk	EcDsScience DataServer	EcStStagingDiskServer	Distributed Object	SDSRV creates Staging Disk for metadata files, which allocates space and passes back a reference to that disk space. The reference to the Staging Disk is determined from the ArchiveID metadata of the granule to be distributed The amount of staging disk to request is determined from collection level metadata from the ESDT's Descriptor file.
B.11.6	Create Metadata file	EcDsScience DataServer			For each granule referenced in the Acquire request, the SDSRV creates a file containing the granule's metadata before passing to Distribution.
B.11.7	Distribute Granules, Synchronous	EcDsScience DataServer	EcDsDdRequestMgrMain	Distributed Object	SDSRV submits a request to DataDistribution. The request includes, for each granule, a reference to the metadata file as well as all data files. Other parameters from the Acquire request are passed to DDIST.

Table 3.3.5.3-1. Component Interaction Table: MODIS Scenario - Standard Production (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.11.8	Create Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST creates Staging Disk for the granule files in the archive. This allocates space and passes back a reference to that disk space. The reference to the Staging Disk is determined from the information passed by the SDSRV in the distribution request ,which was the ArchiveID metadata parameter of the granule to be staged. The amount of staging disk to request is calculated from the file sizes in the information passed in the Distribution Request.
B.11.9	STMGT Retrieve	EcDsDdRequestMgrMain	EcDsStArchiveServer	Distributed Object	DDIST requests that STMGT retrieve the granule file that is archived. This will result in the file being staged to read-only cache disks. This means that all files needed to fulfill the distribution request are on disk, and ready to be copied. The correct archive object to request is determined from the information provided by the SDSRV in the distribution request. This returns references to the files in the read-only cache.
B.11.10	Link files to Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST links the files from the read-only cache into the staging disk.
B.11.11	Copy files to Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST copies the metadata files from the SDSRV's Staging Disk into the staging disk.
B.11.12	ftpPush Files	EcDsDdRequestMgrMain	EcDsStFtpDisServer	Distributed Object	DDIST now creates the Resource manager for ftp Pushes via a Resource Manager Factory. The correct resource manager is determined from the MediaType handed to the resource factory (ftpPush, in this case). The correct FtpServer is determined from configuration within the resource factory. The files, host, location, username and password are all determined from the information provided in the original Acquire request.
B.11.13	ftp Files	EcDsStFtpDisServer	EcDpPrDM	rpc	CSS performs the actual low level ftp of the files.

Table 3.3.5.3-1. Component Interaction Table: MODIS Scenario - Standard Production (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.12.1	Get DSS UR	EcDpPrEM	EcIoAdServer	Distributed Object	If the DSS UR for this Metadata Configuration File (MCF) is not already known in the PDPS database, EM searches the Advertiser for a "GetQueriableParameters" service for the desired MCF data type. This is accomplished via the IoAdApprovedSearchCommand class. Sinthe Advertiser is based on a replicated database, no specification is required to select the proper Advertiser. The local one is used.
B.13.1	Get Metadata Configuration File	EcDpPrEM	EcDsScienceDataServer	Distributed Object	DPS gets the metadata configuration file of the output data's ESDT (MOD02). Datatype and version are from PDPS database; correct client name is from configuration file.
B.13.2	Run PGE	EcDpPrRunPGE	PGE<PGE02>	command line	PGE02is executed. Output MOD02 files are placed in the output directory on Science Hardware disks. The directory path is established by using a root, which was established by configuration, and the specific directory by the job id. This disk root is cross mounted by DPS, SDSRV and STMGT. This is to ensure that they are directly available to the DSS, for archival.
B.14.1	Get DSS UR	EcDpPrDM	EcIoAdServer	Distributed Object	If the DSS UR for this output data type is not already known in the PDPS database, DM searches the Advertiser for a "GetQueriableParameters" service for thesired output data type. This is accomplished via the IoAdApprovedSearchCommand class. Since the Advertiser is based on a replicated database, no specification is required to select the proper Advertiser. The local one is used.
B.15.1	Connect to SDSRV	EcDpPrEM	EcDsScienceDataServer	Distributed Object	PRONG begins a session with the SDSRV by connecting.

Table 3.3.5.3-1. Component Interaction Table: MODIS Scenario - Standard Production (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.15.2	Insert Data	EcDpPrDM	EcDsScienceDataServer	Distributed Object	PRONG requests that the newly created files for the MOD02 granules are inserted into the Data Server. An Insert request, containing the names of the files comprising the granule, is created for each granule. The structure of the Insert Request is hardcoded. SDSRV validates metadata and determines the archived names of the files.
B.15.3	STMGT Store	EcDsScience DataServer	EcDsStArchiveServer	Distributed Object	SDSRV requests that the files are archived. The archive server must be able to read the inserted files directly from the DPS disks that they are residing on. The correct archive object to request is determined from collection level metadata for the ESDT, defined in the ESDT's descriptor.
B.15.4	Adding a Granule to Inventory	EcDsScience DataServer	Sybase/SQS	CtLib	The validated metadata is parsed and added to the inventory of the SDSRV.
B.16.1	Trigger Event	EcDsScience DataServer	EcSbSubServer	Distributed Object	Upon successful insertion of MOD02 granules, the MOD02:Insert event is triggered, for each granule. The correct subscription server is determined from SDSRV configuration. The correct event to trigger are determined from the events file where the event id was stored during the ESDT's installation. Provided with the event triggering is the UR of the inserted granule.
B.16.2	Fire Subscriptions	EcSbSubServer	Sybase	CtLib	SBSRV queries the Sybase database determining which subscriptions need to be activated, or fired. Each query "hit" is an activated subscription and will execute independently.

Table 3.3.5.3-1. Component Interaction Table: MODIS Scenario - Standard Production (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.17.1	Asynchronous Direct Notification	EcSbSubServer	EcPlSubMgr		The SBSRV will notify PLS that there are new MOD02 granules available. The UR of the granule is passed in the notification to the user, along with a reference to the subscription that is being fulfilled. Direct Notification is to a Queuename (See Message Passing Key Mechansim) that PLS- Subscription Manager provided when the subscription was submitted.
B.17.2	Connect to SDSRV	EcPlSubMgr	EcDsScienceDataServer	Distributed Object	Subscription Manager begins a session with the SDSRV by connecting, in order to determine the use of the new granule. The correct SDSRV is determined by using the GranuleUR in the notification message.
B.17.3	Add granules to Session	EcPlSubMgr	EcDsScienceDataServer	Distributed Object	Subscription Manager establishes the data context of the session with the SDSRV by adding new granule's UR. The GranuleUR of the PGE granule is added to the ESDTReferenceCollection.
B.17.4	Retrieve Granule Metadata from Inventory	EcDsScience DataServer	Sybase/SQS	CtLib	SDSRV completes establishing the data context by retrieving the metadata for the requested PGE granule from the Sybase/SQS database. The metadata for the PGE granule is passed back to the reference objects for each granule.
B.17.5	Inspect Granule Value Parameters	EcPlSubMgr	EcDsScienceDataServer	Distributed Object	Subscription Manager checks the new granule's metdata attributes (type, version, filesize and temporal range), to determine which, if any, jobs can use it as input.
B.18.1	Release Job	EcPlSubMgr	EcDpPrJobMgmt	rpc	Planning tells the Job Manager to release the job containing PGE08.
B.18.2	Force Start Job	EcDpPrJobMgmt	event_demon	rpc	Job containing PGE08 is begun.
B.18.3	Job Processing	event_demon	EcDpPrEM	command line	The job containing the PGE08 begins processing.

Table 3.3.5.3-1. Component Interaction Table: MODIS Scenario - Standard Production (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.18.4	Connect to SDSRV	EcDpPrEM	EcDsScienceDataServer	Distributed Object	Processing begins a session with the SDSRV by connecting, in order to acquire the PGE08. The correct SDSRV is determined by using the GranuleUR of the PGE granule, which is defined in the Production plan and is part of the DPR.
B.18.5	Add granules to Session	EcDpPrEM	EcDsScienceDataServer	Distributed Object	PRONG establishes the data context of the session with the SDSRV by adding PGE granule's UR. The GranuleUR of the PGE granule is added to the ESDTReferenceCollection.
B.18.6	Retrieve Granule Metadata from Inventory	EcDsScience DataServer	Sybase/SQS	CtLib	SDSRV completes establishing the data context by retrieving the metadata for the requested PGE granule from the Sybase/SQS database. The metadata for the PGE granule is passed back to the reference objects for each granule.
B.18.7	Acquire Data	EcDpPrEM	EcDsScienceDataServer	Distributed Object	DPS requests granules by submitting an Acquire request for the PGE granule. The Acquire request is for an ftpPush of all granules in the ESDTReferenceCollection. This request is synchronous, meaning that the return of the submit call of the request will contain the results of the request. This means that the response will not be sent until the PGE granule files have been ftp'd to the DPS disks. This request asks for no distribution notice to be email'd. The Acquire request structure is hardcoded.
B.18.8	Create Staging Disk	EcDsScience DataServer	EcStStagingDiskServer	Distributed Object	SDSRV creates Staging Disk for the metadata file, which allocates space and passes back a reference to that disk space. The reference to the Staging Disk is determined from the ArchiveID metadata of the granule to be distributed. The amount of staging disk to request is a default, hardcoded as a SDSRV constant.
B.18.9	Create Metadata file	EcDsScience DataServer			The SDSRV creates a file containing the PGE granule's metadata before passing to Distribution.

Table 3.3.5.3-1. Component Interaction Table: MODIS Scenario - Standard Production (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.18.10	Distribute Granules, Synchronous	EcDsScience DataServer	EcDsDdRequestMgrMain	Distributed Object	SDSRV submits a request to DataDistribution. The request includes, for each granule, a reference to the metadata file as well as all data files. Other parameters from the Acquire request are passed to DDIST.
B.18.11	Create Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST creates Staging Disk for the granule files in the archive. This allocates space and passes back a reference to that disk space. The reference to the Staging Disk is determined from the information passed by the SDSRV in the distribution request ,which was the ArchiveID metadata parameter of the granule to be staged. The amount of staging disk to request is calculated from the file sizes of archived files, in the information passed in the Distribution Request.
B.18.12	STMGT Retrieve	EcDsDdRequestMgrMain	EcDsStArchiveServer	Distributed Object	DDIST requests that STMGT retrieve the PGE granule file that is archived. This will result in the file being staged to read-only cache disks. This means that all files needed to fulfill the distribution request are on disk, and ready to be copied. The correct archive object to request is determined from the information provided by the SDSRV in the distribution request. This returns references to the files in the read-only cache.
B.18.13	Link files to Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST links the files from the read-only cache into the staging disk.
B.18.14	Copy files to Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST copies the metadata files from the SDSRV's Staging Disk into the staging disk.

Table 3.3.5.3-1. Component Interaction Table: MODIS Scenario - Standard Production (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.18.15	ftpPush Files	EcDsDdRequestMgrMain	EcDsStFtpDisServer	Distributed Object	DDIST now creates the Resource manager for ftp Pushes via a Resource Manager Factory. The correct resource manager is determined from the MediaType handed to the resource factory (ftpPush, in this case). The correct FtpServer is determined from configuration within the resource factory. The files, host, location, username and password are all determined from the information provided in the original Acquire request.
B.18.16	ftp Files	EcDsStFtpDisServer	EcDpPrEM	rpc	CSS performs the actual low level ftp of the PGE files.
B.19.1	Get DSS RU	EcDpPrEM	EcIoAdServer	Distributed Object	If the DSS UR for this Metadata Configuration File (MCF) is not already known in the PDPS database, EM searches the Advertiser for a "GetQueriableParameters" service for the desired MCF data type. This is accomplished via the IoAdApprovedSearchCommand class. Sinthe Advertiser is based on a replicated database, no specification is required to select the proper Advertiser. The local one is used.
B.20.1	Get Metadata Configuration File	EcDpPrEM	EcDsScienceDataServer	Distributed Object	DPS gets the metadata configuration file of the output data's ESDT (MOD029). Datatype and version are from PDPS database; correct client name is from configuration file.
B.20.2	Run PGE	EcDpPrRunPGE	PGE<PGE08>		PGE08 is executed. Output MOD029 files are placed in the output directory on Science Hardware disks. The directory path is established by using a root, which was established by configuration, and the specific directory by the job id. This disk root is cross mounted by DPS, SDSRV and STMGT. This is to ensure that they are directly available to the DSS, for archival.
B.21.1	Detecting a Failed PGE	EcDpPrEM			The log file generated by ECDpPrRunPGE is inspected for failure exit codes. This processing continues with Thread C of the MODIS scenario.

Table 3.3.5.3-1. Component Interaction Table: MODIS Scenario - Standard Production (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.22.1	Get DSS UR	EcDpPrDM	EcIoAdServer	Distributed Object	If the DSS UR for this output data type is not already known in the PDPS database, DM searches the Advertiser for a "GetQueriableParameters" service for thesired output data type. This is accomplished via the IoAdApprovedSearchCommand class. Since the Advertiser is based on a replicated database, no specification is required to select the proper Advertiser. The local one is used.
B.23.1	Connect to SDSRV	EcDpPrEM	EcDsScienceDataServer	Distributed Object	PRONG begins a session with the SDSRV by connecting.
B.23.2	Insert Data	EcDpPrEM	EcDsScienceDataServer	Distributed Object	PRONG requests that the newly created files for the MOD029 granule are inserted into the Data Server. An Insert request, containing the names of the files comprising the granule, is created for each granule. The structure of the Insert Request is hardcoded. SDSRV validates metadata and determines the archived names of the files.
B.23.3	STMGT Store	EcDsScience DataServer	EcDsStArchiveServer	Distributed Object	SDSRV requests that the files are archived. The archive server must be able to read the inserted files directly from the DPS disks that they are residing on. The correct archive object to request is determined from collection level metadata for the ESDT, defined in the ESDT's descriptor.
B.23.4	Adding a Granule to Inventory	EcDsScience DataServer	Sybase/SQS	CtLib	The validated metadata is parsed and added to the inventory of the SDSRV.
B.24.1	Trigger Event	EcDsScience DataServer	EcSbSubServer	Distributed Object	Upon successful insertion of MOD029 granule, the MOD029:Insert event is triggered. The correct subscription server is determined from SDSRV configuration. The correct events to trigger are determined from the events file, where the event id was stored during ESDT installation. Provided with the event triggering is the UR of the inserted granule.

Table 3.3.5.3-1. Component Interaction Table: MODIS Scenario - Standard Production (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.24.2	Fire Subscriptions	EcSbSubServer	Sybase	CtLib	SBSRV queries the Sybase database determining which subscriptions need to be activated, or fired. Each query "hit" is an activated subscription and will execute independently.

3.3.6 Failed PGE Handling Thread

3.3.6.1 Interaction Diagram - Domain View

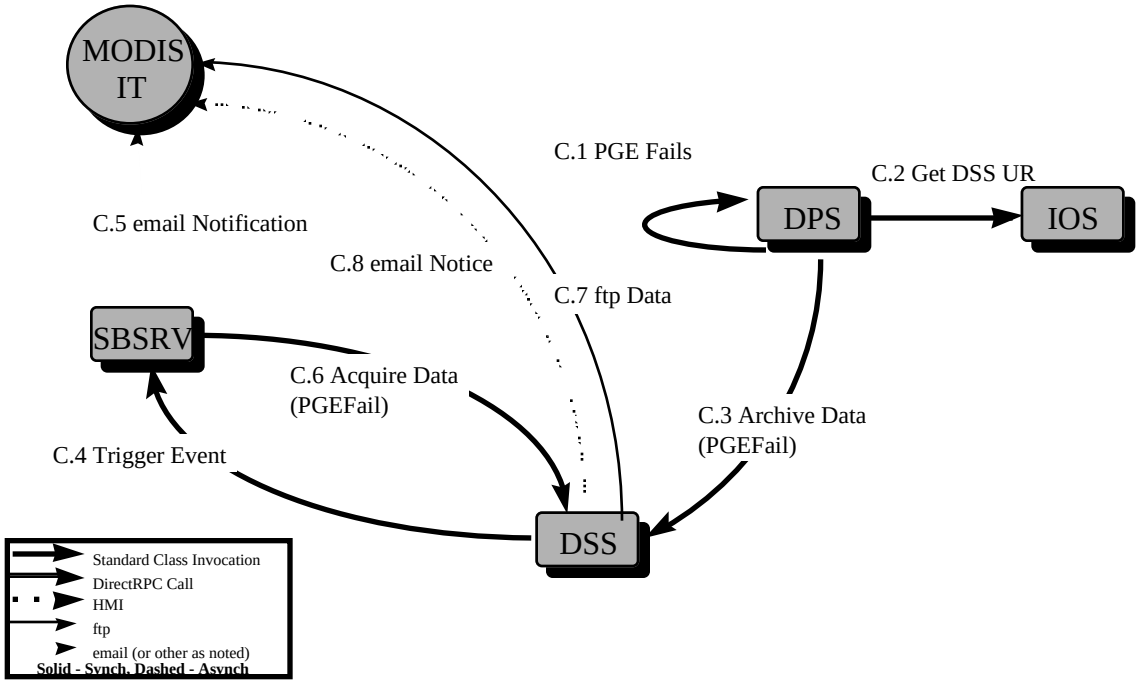

Figure 3.3.6.1-1. Failed PGE Handling Interaction Diagram

3.3.6.2 Interaction Table - Domain View

Table 3.3.6.2-1. Interaction Table - Domain View: MODIS Scenario, Failed PGE Handling

Step	Interaction	Interface Client	Interface Provider	Data Issues	Preconditions	Description
C.1	PGE Fails	DPS	PGE			One instance of the PGE08 running fails, due to the need for all daylight data, but some input data is night data. This step is the same step as B.14 in the previous Thread.
C.2	Get DSS UR	DPS	IOS			DPS gets the DSS UR from Advertiser.
C.3	Archive Data	DPS	DSS	1 PGEFail @30K	PGEFail	DPS collects the artifacts from the failed PGE, tar's and inserts them into the Data Server.
C.4	Trigger Event	DSS	SBSRV			Trigger FailPGE:Insert event upon successful insertion of the FailPGE granule.
C.5	Notification	SBSRV	MODIS Instrument Team		MODIS IT Subscription for FailPGE:Insert event, qualified for MODIS PGEs.	Send email notification to MODIS IT, notifying that there is a newly inserted FailPGE granule, from a MODIS PGE. Notification message includes the UR of the FailPGE granule.
C.6	Acquire Data	SBSRV	DSS			SBSRV fulfills the standing order by the MODIS IT, for Failed MODIS PGEs. Request to Acquire data, via ftpPush, to the MODIS IT's host.
C.7	ftp Data	DSS	MODIS Instrument Team			Data Server ftp's the FailPGE tar file to the MODIS IT, placing it in the specified directory on the specified host.
C.8	Distribution Notice	DSS	MODIS Instrument Team			Send email notification to MODIS IT, notifying that the newly inserted FailPGE has been successfully ftp'd to their machine.

3.3.6.3 Component Interaction Table

Table 3.3.6.3-1. Component Interaction Table: MODIS Scenario, Failed PGE Handling

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
C.1.1	Detecting a Failed PGE	DPRExecution	PGE	File containing exit code	One instance of the PGE08 has failed, due to incorrect input data. This is detected by examining the exit code of the PGE, which is stored in a file named <DPRID>.log (in this case, MODPGE08#1.0#01<time>.log). DPS pulls together core file, along with any other files marked in the PCF, and tar's them together. Metadata for the PGEFail is built by EcDpPrPGE process.
C.2.1	Get DSS UR	EcDpPrDM	EcIoAdServer	Distributed Object	If the DSS UR for this output data type is not already known in the PDPS database, DM searches the Advertiser for a "GetQueriableParameters" service for thesired output data type. This is accomplished via the IoAdApprovedSearchCommand class. Since the Advertiser is based on a replicated database, no specification is required to select the proper Advertiser. The local one is used.
C.3.1	Connect to SDSRV	DPRExecution	EcDsScience DataServer	Distributed Object	DPRExecution begins a session with the SDSRV by connecting.
C.3.2	Insert Data	DPRExecution	EcDsScience DataServer	Distributed Object	PRONG requests that the newly created tar file for the PGEFail granule are inserted into the Data Server. An Insert request, containing the names of the file comprising the granule, is created for the granule. The structure of the Insert Request is hardcoded. SDSRV validates metadata and determines the archived names of the files.
C.3.3	STMGT Store	EcDsScienceDataServer	EcDsStArchiveServer	Distributed Object	SDSRV requests that the files are archived. The archive server must be able to read the inserted files directly from the DPS disks that they are residing on. The correct archive object to request is determined from collection level metadata for the FailPGE ESDT, defined in the ESDT's descriptor.
C.3.4	Adding a Granule to Inventory	EcDsScienceDataServer	Sybase/SQS	CtLib	The validated metadata is parsed and added to the inventory of the SDSRV.

Table 3.3.6.3-1. Component Interaction Table: MODIS Scenario, Failed PGE Handling (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
C.4.1	Trigger Event	EcDsScienceDataServer	EcSbSubServer	Distributed Object	Upon successful insertion of PGEFail granule, the PGEFail:Insert event is triggered. The correct subscription server is determined from SDSRV configuration. The correct events to trigger are determined from the events file which was populated during ESDT installation. Provided with the event triggering is the UR of the inserted granule.
C.4.2	Fire Subscriptions	EcSbSubServer	Sybase	CtLib	SBSRV queries the Sybase database determining which subscriptions need to be activated, or fired. Each query “hit” is an activated subscription and will execute independently.
C.5.1	Send E-mail	EcSbSubServer	MODIS IT		The SBSRV builds an email notification that the user’s subscription on the PGEFail event has been fired. This notification will identify the Event, the subscription ID, the granule UR that was inserted and the previously supplied UserString.
C.6.1	Connect to SDSRV	EcSbSubServer	EcDsScience DataServer	Distributed Object	In order to fulfill a standing order, the SBSRV begins a session with the SDSRV, on behalf of the subscription user. The correct SDSRV is determined from the Granule UR provided with the event triggering.
C.6.2	Add granules to Session	EcSbSubServer	EcDsScience DataServer	Distributed Object	The SBSRV establishes the data context of the session with the SDSRV by adding the input granules to the session. The GranuleUR of each input granule is added to the ESDTReferenceCollection.
C.6.3	Retrieve Granule Metadata from Inventory	EcDsScienceDataServer	Sybase/SQS	CtLib	SDSRV completes establishing the data context by retrieving the metadata for the requested granules from the Sybase/SQS database. The metadata for each granule is passed back to the reference objects for each granule.
C.6.4	Acquire Data	EcSbSubServer	EcDsScience DataServer	Distributed Object	SBSRV fulfills the standing order for the PGEFail granule by submitting an Acquire request for the granule. The Acquire request is for an ftpPush of all granules in the ESDTReferenceCollection. This request is asynchronous, meaning that the return of the submit call of the request will only contain the status of the request’s submittal. This request asks for a distribution notice to be email’d to the client. The Acquire request structure was hardcoded within the subscription server.

Table 3.3.6.3-1. Component Interaction Table: MODIS Scenario, Failed PGE Handling (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
C.6.5	Create Staging Disk	EcDsScienceDataServer	EcStStagingDiskServer	Distributed Object	SDSRV creates Staging Disk for metadata files, which allocates space and passes back a reference to that disk space. The reference to the Staging Disk is determined from the ArchiveID metadata of the granule to be distributed The amount of staging disk to request is determined from collection level metadata from the ESDT's Descriptor file.
C.6.6	Create Metadata file	EcDsScienceDataServer			For each granule referenced in the Acquire request, the SDSRV creates a file containing the granule's metadata before passing to Distribution.
C.6.7	Distribute Granules, Synchronous	EcDsScienceDataServer	EcDsDdRequestMgrMain	Distributed Object	SDSRV submits a request to Data Distribution. The request includes, for the granule, a reference to the metadata file as well as the data file. Other parameters from the Acquire request are passed to DDIST.
C.6.8	Create Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST creates Staging Disk for the granule files in the archive. This allocates space and passes back a reference to that disk space. The reference to the Staging Disk is determined from the information passed by the SDSRV in the distribution request ,which was the ArchiveID metadata parameter of the granule to be staged. The amount of staging disk to request is calculated from the file sizes in the information passed in the Distribution Request.
C.6.9	STMGT Retrieve	EcDsDdRequestMgrMain	EcDsStArchiveServer	Distributed Object	DDIST requests that STMGT retrieve the granule file that is archived. This will result in the file being staged to read-only cache disks. This means that all files needed to fulfill the distribution request are on disk, and ready to be copied. The correct archive object to request is determined from the information provided by the SDSRV in the distribution request. This returns references to the files in the read-only cache.
C.6.10	Link files to Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST links the files from the read-only cache into the staging disk.
C.6.11	Copy files to Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST copies the metadata files from the SDSRV's Staging Disk into the staging disk.

Table 3.3.6.3-1. Component Interaction Table: MODIS Scenario, Failed PGE Handling (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
C.6.12	ftpPush Files	EcDsDdRequestMgrMain	EcDsStFtpDisServer	Distributed Object	DDIST now creates the Resource manager for ftp Pushes via a Resource Manager Factory. The correct resource manager is determined from the MediaType handed to the resource factory (ftpPush, in this case). The correct FtpServer is determined from configuration within the resource factory. The files, host, location, username and password are all determined from the information provided in the original Acquire request.
C.7.1	ftp Files	EcDsStFtpDisServer	MODIS IT	rpc	CSS performs the actual low level ftp of the files.
C.8.1	Build Distribution Notice	EcDsDdRequestMgrMain			The DDIST builds an email notification that the user's order has been fulfilled. This notification will include the media id, type and format, as well as UR, type and file names and sizes for each granule.
C.8.2	Send E-mail	EcDsDdRequestMgrMain	MODIS IT		DDIST sends the distribution notice to the user via email. Since Standing Orders are not currently tracked orders, and the user DDIST sends the notification to is determined from the Order, the notice is currently sent to a pre-configured default Email address, for DAAC Distribution Technician parsing and forwarding.

3.3.7 Data Access Thread

3.3.7.1 Interaction Diagram - Domain View

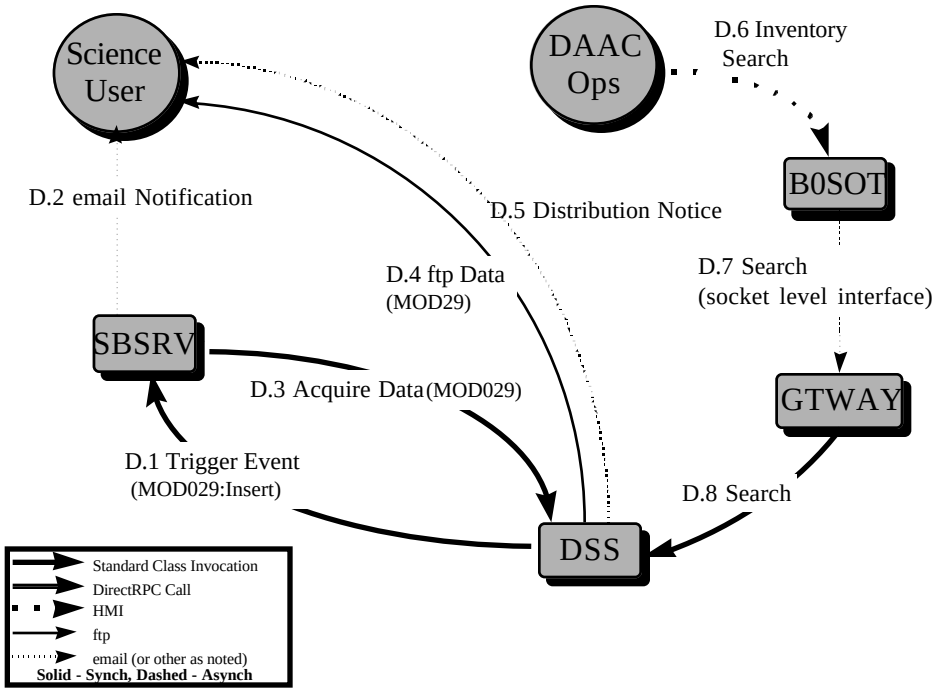

Figure 3.3.7.1-1. Data Access Interaction Diagram

3.3.7.2 Interaction Table - Domain View

Table 3.3.7.2-1. Interaction Table - Domain View: MODIS Scenario, Data Access

Step	Interaction	Interface Client	Interface Provider	Data Issues	Preconditions	Description
D.1	Trigger Event	DSS	SBSRV			This thread picks up with fulfilling the standing order for MOD029 data. This is the same step as B.19 of this MODIS scenario.
D.2	Notification	SBSRV	Science User			Send email notification to Science User, notifying that there is a newly inserted MOD029 granule. Notification message includes the UR of the MOD029 granule.

Step	Interaction	Interface Client	Interface Provider	Data Issues	Preconditions	Description
D.3	Acquire Data	SBSRV	DSS			SBSRV fulfills the standing order by the Science User, for MOD029 granules. Request to Acquire data, via ftpPush, to the Science User's host.
D.4	ftp Data	DSS	Science User	1 MOD029 @ 10.9MB		Data Server ftp's the MOD029 granule's files to the Science User, placing them in the specified directory on the specified host.
D.5	Distribution Notice	DSS	Science User			Send email notification to Science User, notifying that the newly inserted MOD029 granule has been successfully ftp'd to their machine.
D.6	Inventory Search	DAAC Operator	B0SOT			In order to verify that the newly created data is available, a DAAC Operator will perform an inventory search for all MODIS data created in the last day.
D.7	Search	B0SOT	GTWAY			B0SOT submits the DAAC Operator's search criteria to the V0 Gateway in ODL format, via a well defined socket.
D.8	Search	GTWAY	DSS			The V0 gateway translates the Search criteria from ODL to a query object (using GIParameters), and submits that query to the Search service. The results of this Search are returned synchronously, and are passed back to B0SOT, which displays them to the DAAC Operator.

3.3.7.3 Component Interaction Table

Table 3.3.7.3-1. Component Interaction Table: MODIS Scenario, Data Access

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
D.1.1	Trigger Event	EcDsScienceDataServer	EcSbSubServer	Distributed Object	Upon successful insertion of MOD029 granules, the MOD029:Insert event is triggered, for each granule. The correct subscription server is determined from SDSRV configuration. The correct events to trigger are determined from the events file, which was populated during ESDT installation,.. Provided with the event triggering is the UR of the inserted granule.
D.1.2	Fire Subscriptions	EcSbSubServer	Sybase	CtLib	SBSRV queries the Sybase database determining which subscriptions need to be activated, or fired. Each query "hit" is an activated subscription and will execute independently.
D.2.1	Send E-mail	EcSbSubServer	Science User		The SBSRV builds an email notification that the user's subscription on the MOD029:Insert event has been fired. This notification will identify the Event, the subscription ID, the GranuleUR that was inserted and the previously supplied UserString.
D.3.1	Connect to SDSRV	EcSbSubServer	EcDsScienceDataServer	Distributed Object	In order to fulfill a standing order for the MOD029 data, the SBSRV begins a session with the SDSRV, on behalf of the subscription user. The correct SDSRV is determined from the Granule UR provided with the event triggering.
D.3.2	Add granules to Session	EcSbSubServer	EcDsScienceDataServer	Distributed Object	The SBSRV establishes the data context of the session with the SDSRV by adding the input granules to the session. The GranuleUR of each input granule is added to the ESDTReferenceCollection.
D.3.3	Retrieve Granule Metadata from Inventory	EcDsScienceDataServer	Sybase/SQS	CtLib	SDSRV completes establishing the data context by retrieving the metadata for the requested granules from the Sybase/SQS database. The metadata for each granule is passed back to the reference objects for each granule.
D.3.4	Acquire Data	EcSbSubServer	EcDsScienceDataServer	Distributed Object	SBSRV fulfills the standing order for the MOD029 granule by submitting an Acquire request for the granule. The Acquire request is for an ftpPush of all granules in the ESDTReferenceCollection. This request is asynchronous, meaning that the return of the submit call of the request will only contain the status of the request's submittal. This request asks for a distribution notice to be email'd to the client. The Acquire request structure was determined from the Action submitted with the standing order method.

Table 3.3.7.3-1. Component Interaction Table: MODIS Scenario, Data Access (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
D.3.5	Create Staging Disk	EcDsScienceDataServer	EcStStagingDiskServer	Distributed Object	SDSRV creates Staging Disk for metadata files, which allocates space and passes back a reference to that disk space. The reference to the Staging Disk is determined from the ArchiveID metadata of the granule to be distributed The amount of staging disk to request is determined from collection level metadata from the ESDT's Descriptor file.
D.3..6	Create Metadata file	EcDsScienceDataServer			For each granule referenced in the Acquire request, the SDSRV creates a file containing the granule's metadata before passing to Distribution.
D.3.7	Distribute Granules, Synchronous	EcDsScienceDataServer	EcDsDdRequestMgrMain	Distributed Object	SDSRV submits a request to Data Distribution. The request includes, for the granule, a reference to the metadata file as well as the data file. Other parameters from the Acquire request are passed to DDIST.
D.3.8	Create Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST creates Staging Disk for the granule files in the archive. This allocates space and passes back a reference to that disk space. The reference to the Staging Disk is determined from the information passed by the SDSRV in the distribution request ,which was the ArchiveID metadata parameter of the granule to be staged. The amount of staging disk to request is calculated from the file sizes in the information passed in the Distribution Request.
D.3.9	STMGT Retrieve	EcDsDdRequestMgrMain	EcDsStArchiveServer	Distributed Object	DDIST requests that STMGT retrieve the granule file that is archived. This will result in the file being staged to read-only cache disks. This means that all files needed to fulfill the distribution request are on disk, and ready to be copied. The correct archive object to request is determined from the information provided by the SDSRV in the distribution request. This returns references to the files in the read-only cache.
D.3.10	Link files to Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST links the files from the read-only cache into the staging disk.
D.3.11	Copy files to Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST copies the metadata files from the SDSRV's Staging Disk into the staging disk.

Table 3.3.7.3-1. Component Interaction Table: MODIS Scenario, Data Access (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
D.3.12	ftpPush Files	EcDsDdRequestMgrMain	EcDsStFtpDisServer	Distributed Object	DDIST now creates the Resource manager for ftp Pushes via a Resource Manager Factory. The correct resource manager is determined from the MediaType handed to the resource factory (ftpPush, in this case). The correct FtpServer is determined from configuration within the resource factory. The files, host, location, username and password are all determined from the information provided in the original Acquire request.
D.4.1	ftp Files	EcDsStFtpDisServer	Science User	rpc	CSS performs the actual low level ftp of the files.
D.5.1	Build Distribution Notice	EcDsDdRequestMgrMain			The DDIST builds an email notification that the user's order has been fulfilled. This notification will include the media id, type and format, as well as UR, type and file names and sizes for each granule.
D.5.2	Send E-mail	EcDsDdRequestMgrMain	Science User		DDIST sends the distribution notice to the user via email. Since Standing Orders are not currently tracked orders, and the user DDIST sends the notification to is determined from the Order, the notice is currently sent to a pre-configured default Email address, for DAAC Distribution Technician parsing and forwarding.
D.6.1	Startup B0SOT	DAAC Science Data Specialist	xims	XEvent	DAAC Science Data Specialist invokes the B0SOT GUI Application. The operator has already been logged on the DAAC Desktop and begins B0SOT by double-clicking on an icon.
D.6.2	Select Inventory Search, Provide Query constraints, Submit Query	DAAC Ops	xims	GUI	The operator provides search constraints and the products desired. When query constraints are completed, the query is submitted.
D.7.1	V0 Gateway Inventory	xims	EcDmV0ToEcsGateway	ODL, over sockets	B0SOT submits a search to the V0 Gateway, by converting the search criteria into an ODL structure and passing that structure to a socket provided by the Gateway. The correct socket is determined from configuration information contained in the Valids file.
D.8.1	Establish ECS User	EcDmV0ToEcsGateway	MsAcManager	Distributed Object	V0 Gateway retrieves the User Profile using ECS Authenticator from ODL message, which includes an encrypted UserID and Password. The User Registration Server is replicated across DAACs, so the connection is made to the local User Registration Server.

Table 3.3.7.3-1. Component Interaction Table: MODIS Scenario, Data Access (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
D.8.2	Translate Terms	EcDmV0ToEcsGateway	EcDmDictServer	Ctlib (RWDbTool)	Gateway translates the V0 terms from ODL into ECS names for quirey submittal. Interface currently is directly to Data Dictionary database. Database name is retrieved from configuration file.
D.8.3	Connect to SDSRV	EcDmV0ToEcsGateway	EcDsScienceDataServer	Distributed Object	The Gateway first connects to the SDSRV. The correct SDSRV is determined by configuration information.
D.8.4	SDSRV Query	EcDmV0ToEcsGateway	EcDsScienceDataServer	Distributed Object	The Gateway translates the query into a DsClQuery object. This object is handed to the Search interface of the DsClESDTReferenceCollector. This Search method is synchronous, so the results of the search will be returned to the calling function.
D.8.5	Querying the Inventory	EcDsScienceDataServer	Sybase/SQS	CtLib	The SDSRV breaks down the Query object and translates it into a sequence of calls to the inventory database. Resultant rows are converted into data granules, each with their metadata extracted from the database. These results are packaged and returned to the Query client.
D.8.6	Result Retrieval	xims	EcDmV0ToEcsGateway	ODL, over Sockets	When the Gateway gets the results, they are translated into ODL, and passed back to the B0SOT tool. The correct socket for sending results to B0SOT is the one used to submit the query. B0SOT then displays the results of the query to the user.

3.4 Landsat Scenario

3.4.1 Scenario Description

This scenario shows how ECS supports the archival and retrieval of Landsat 7 data. Landsat 7 data is provided to ECS from one of two primary sources: LPS and IAS. LPS provides Landsat L0R subinterval data to ECS. IAS provides the Landsat Calibration Parameter files to ECS. The dialogue between LPS and ECS is managed by the Landsat 7 Gateway. The interface between IAS and ECS is through a standard polling mechanism managed by Ingest. The dialogue and polling mechanism are documented in the ECS-Landsat 7 ICD[2] . ECS does not process Landsat 7 data.

[2] ESDIS Document, 505-41-32, Interface Control Document between the EOSDIS Core System (ECS) and the Landsat 7 System, Revision A. May 1997

There are some notable data issues related to the Landsat 7 L0R subintervals. Actually, the L0R data is produced by LPS in two independent parts, Format 1 and Format 2. While both parts are required to make a complete L0R granule, these parts are provided to ECS independent from each other. Additionally, there is no guarantee of the delivery order of the parts: Format one could precede or follow the delivery of its corresponding Format two. The deliveries might be separated by delivery of parts of other granules. This requires ECS to determine if each received part's corresponding part has already been delivered. If so, further internal processing is performed to match the parts and to create the complete L0R subinterval granule.

Another point of interest is that while LPS provides ECS with all L0R data, which is subinterval data, ECS provides access to standard WRS Scenes within the subinterval data. When LPS provides a subinterval, all WRS scene data is embedded in the subinterval and metadata is provided to derive those scenes. The current ECS approach to storing and accessing subintervals and scenes is to store all subinterval data, and to derive scenes from the subinterval when ECS users request the scenes. To support this two things occur: First when subinterval data is entered into the Data Server, references to all scenes that are available within that subinterval are also created. These scenes are "virtual granules", insofar as they are referenced as granules, and will be searchable, but the granule itself is not archived, per se. Rather it is created upon access. Secondly, upon access the scenes are derived from their "parent" subinterval by using internal subinterval subsetting services. Effectively, each scene knows which subinterval it is part of, and it knows what to ask its subinterval to do in order to complete creation of itself.

This scenario does not address some of the special circumstances required by ECS to support the Landsat mission. Most notably, this includes Landsat costing, estimating and billing. There are workarounds established, but they were not exercised in this scenario.

The following system functionality is exercised in this scenario:

- User Profile Registration
- LPS-driven data ingest
- IAS-driven data ingest
- Access to WRS Scenes
- B0SOT browse of granules
- 8mm data distribution
- Operator ability to status an order.

3.4.2 Scenario Preconditions

The following ESDTs have been inserted into the ECS:

L70R (Complete L7 L0R subinterval)
L70RF1 (Format 1 of a L0R subinterval)
L70RF2 (Format 2 of a L0R subinterval)
L70RWRS (Complete L7 WRS Scene)
L70RWRS1 (Format 1 part of a WRS Scene)
L70RWRS2 (Format 2 part of a WRS Scene)
L7CPF (Landsat 7 Calibration Parameter File)

Subscription for Landsat 7 subintervals entered on behalf of Instrument Team.

Standing order for WRS Scene, to be delivered on 8mm tape.

3.4.3 Scenario Partitions

The Landsat scenario has been partitioned into the following threads:

User Registration (Thread A) - This thread will show the processing required for registering as a new ECS User.

LPS Data Insertion (Thread B) - This thread will show how the ECS inserts data provided by LPS.

Standing Order Support (Thread C) - This thread will show how the system supports standing orders for data granules.

IAS Data Insertion (Thread D) - This thread will show how the ECS inserts data provided by IAS.

Search and Browse (Thread E) - This thread will show how the ECS supports users searching for and browsing data holdings.

Ordering WRS Scenes (Thread F) - This thread will show how the ECS supports user orders for WRS scenes.

3.4.4 User Registration Thread

3.4.4.1 Interaction Diagram - Domain View

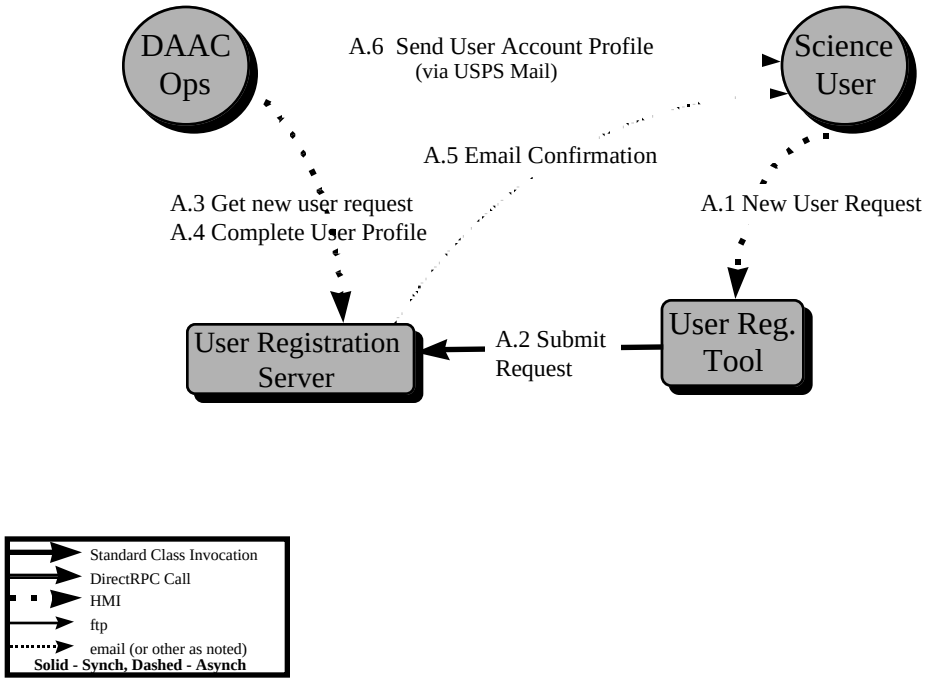

Figure 3.4.4.1-1. User Registration Interaction Diagram

3.4.4.2 Interaction Table - Domain View

Table 3.4.4.2-1. Interaction Table - Domain View: Landsat 7 Scenario, User Registration

Step	Interaction	Interface Client	Interface Provider	Data Issues	Preconditions	Description
A.1	New User Request	Science User	User Registration Tool		Science User is running within the Science Desktop	Science user loads User Registration Tool, via its URL, from their favorite Web Browser. Science user fills out form with initial registration information. This information includes: username, address, telephone number, email address and mother's maiden name (for security confirmation). Request is queued at DAAC.
A.2	Submit Request	User Registration Tool	User Registration Server			User Registration Tool submits the new user's request. The request is queued at the DAAC, awaiting the DAAC User Services staff to confirm the new user.
A.3	Get New User Request	DAAC User Services Representative	User Registration Server			DAAC User Services Representative (periodically) checks for new user registration requests. In this case the request for our new user is found. User Services staff checks the information provided.
A.4	Complete User Profile	DAAC User Services Representative	User Registration Server			DAAC User Services Representative completes the new user's User Profile. The request is marked as confirmed and accepted. DAAC User Services Representative may call Science User for any further information or clarification.
A.5	Email Confirmation	User Registration Server	Science User			User Registration Server emails out confirmation of the user's registration request.
A.6	Mail User Account Profile	DAAC User Services Representative	Science User			DAAC User Services Representative sends complete user account profile, including passwords, to Science User via USPS mail.

3.4.4.3 Component Interaction Table

Table 3.4.4.3-1. Component Interaction Table: Landsat Scenario, User Registration

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
A.1.1	Startup User Registration Tool	Science User	EcClDtUser ProfileGate way	Web Browser	Science User, double clicks on icon for user registration, which is standard on the Science Desktop. This invokes the configured Web Browser, with the URL of the User Registration Tool.
A.1.2	Input User Registration Information	Science User	EcClDtUser ProfileGate way	Web Browser	The Science User populates forms with ECS registration information. This information includes: username, address, telephone number, email address and mother's maiden name (for security confirmation). The user then submits this information.
A.2.1	Submit User Registration Request	EcClDtUser ProfileGate way	MsAcMana ger	Distribute d Object	The User Registration Tool submits the User Registration Request to the User Registration Server for approval.
A.2.2	Persist a User Registration Request	MsAcMana ger	Sybase	CtLib	The User Registration Request is saved for approval by DAAC User Servics.
A.3.1	Startup User Registration Server GUI	DAAC Ops - User Services	MsAcRegU serGUI	Xterm	DAAC operations starts the User Registration Server GUI. There is an icon on the DAAC Desktop that represents the GUI process.
A.3.2	Review New User Request	DAAC Ops - User Services	MsAcRegU serGUI	Xterm	On a periodic basis (based on DAAC policy), User Services checks for any new User Registration Requests.
A.3.3	Get New User Requests	MsAcRegU serGUI	MsAcMana ger	Distribute d Object	Request all new User Registration Requests. The GUI connects to the Registration Server that is local to the DAAC. This is determined by a CDS entry in the GUI's configuration.
A.3.4	Retrieve User Registration Requests	MsAcMana ger	Sybase	CtLib	All pending User Registration Request are retrieved from the database.
A.4.1	Update User Profile	MsAcRegU serGUI	MsAcMana ger	Distribute d Object	DAAC User Services completes the User Profile from the request. Updated information includes: DCE Username, group and password, V0Gateway Userrname, group and password.
A.4.2	Create User Profile	MsAcMana ger			User Registration Server takes the completed User Registration Request and makes a User Profile, registering the user.
A.4.3	Persist a UserProfile	MsAcMana ger	Sybase	CtLib	The User Profile is saved in the system.
A.5.1	Build User Confirmation	MsAcMana ger			The User Registration Server builds a new user confirmation message to be sent. Username and address information is included in this message.
A.5.2	Send E-mail	MsAcMana ger	Science User		The Confirmation message is sent to the new ECS Science User, via CSS infrastructure mail services.

3.4.5 LPS Data Insertion Thread

3.4.5.1 Interaction Diagram - Domain View

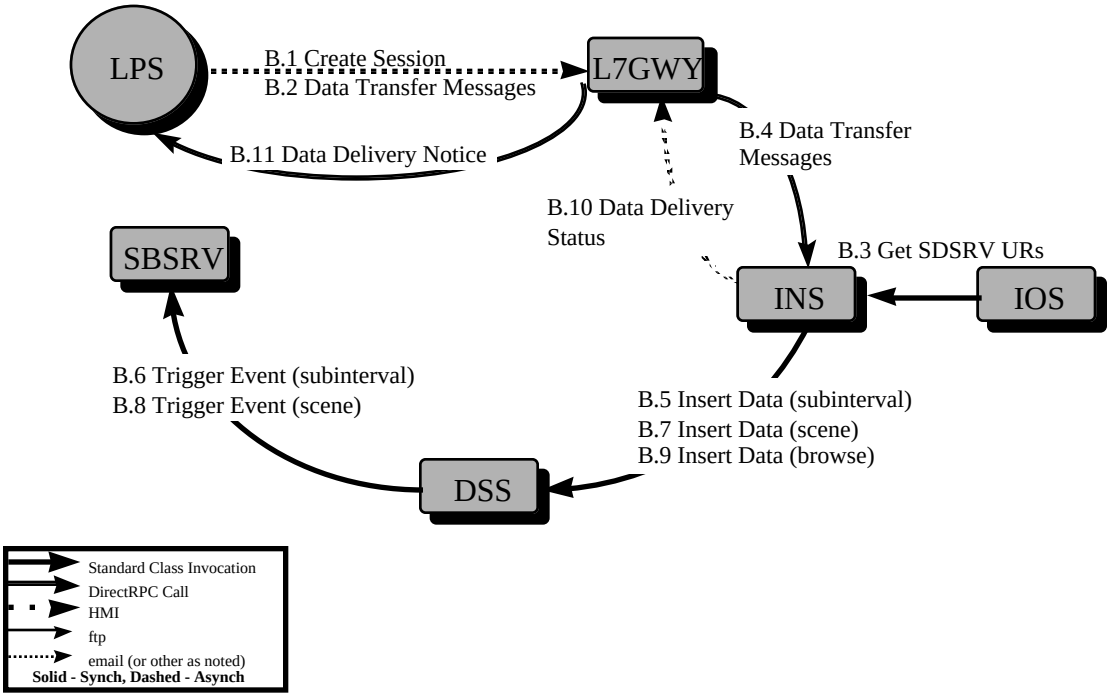

Figure 3.4.5.1-1. LPS Data Insertion Interaction Diagram

3.4.5.2 Interaction Table - Domain View

Table 3.4.5.2-1. Interaction Table - Domain View: Landsat 7 Scenario, LPS Data Insertion

Step	Interaction	Interface Client	Interface Provider	Data Issues	Preconditions	Description
B.1	Create Session	LPS	L7GWY		All required binding information is provided to LPS.	In order to transfer L7 subinterval data, LPS initiates a session with ECS. This is accomplished by sending a series of control messages to the L7GWY.
B.2	Data Transfer Messages	LPS	L7GWY			Once a session is established with ECS, LPS begins transfer of the L7 data through a series of messages, beginning with a Data Availability Notice (DAN).
B.3	Get SDSRV Urs	INS	IOS			Upon startup, Ingest gets the SDSRV Urs for each data type in its database.
B.4	Data Transfer Messages	L7GWY	INS			L7GWY passes all Data Transfer messages to the Ingest subsystem.
B.5	Insert Data	INS	DSS	13 L70RF1 files @ 732MB, or 7 L70RF2 @ 387MB, from Landsat Team	L70RF1, L70RF2 and L70R ESDTs.	When complete L70R data is transferred to Ingest disk, Ingest sends the data to DSS for archival.
B.6	Trigger Event	DSS	SBSRV			Upon successful completion of insertion of L70RF1, L70RF2, or complete L70R subinterval, the appropriate Insert event is triggered.
B.7	Insert Data	INS	DSS	3 L70RWRS(F1/F2) virtual granules		Ingest derives WRS scene information from the subinterval metadata. Virtual scene granules (metadata only) are inserted into the Data Server.
B.8	Trigger Event	DSS	SBSRV			Data server triggers the scene's :Insert event when the scene is successfully saved.
B.8	Insert Data	INS	DSS	3 (one per scene) browse granule @500K		Ingest derives browse data from the subinterval. Ingest then inserts the browse into the Data Server.
B 10	Data Delivery Status	INS	L7GWY			Ingest create status acknowledgement message to return to LPS.
B.11	Data Delivery Notice	L7GWY	LPS			L7 Gateway passes Data Delivery Notice to LPS.

3.4.5.3 Component Interaction Table

Table 3.4.5.3-1. Component Interaction Table: Landsat Scenario, LPS Data Insertion

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.1.1	LPS Create Session	LPS	EcCsLandsat7Gateway	rpc	LPS and the Gateway server exchange messages to begin a session. The Session messages include Authentication Request and Authentication Response, which are documented in the ECS - Landsat 7 ICD.
B.2.1	Begin LPS Data transfer	LPS	EcCsLandsat7Gateway	rpc	LPS first sends a DAN, letting ECS know that data is available, what data it is and where it is. ECS will respond with a DAA. After insertion, ECS provides a DDN, which will be acknowledged by LPS with a DDA.
B.3.1	Get SDSRV URs fromIOS	EcInReqMgr	EcIoAdServer	Distributed Object	Upon startup, Ingest Request Manager requests the SDSRV URs for each data type in its database.
B.4.1	Data Transfer Messages	EcCsLandsat7Gateway	EcInAuto	Distributed Object	The L7 Gateway transfers the LPS data exchange messages to Ingest. Ingest provides responses in accordance with the L7 - ECS ICD.
B.4.2	Ingest Request	EcInAuto	EcInReqMgr	Distributed Object	Auto Ingest process packages the Data Transfer messages into the appropriate Ingest Requests. The data source (LPS), defined on startup, is passed to the Ingest Request Manager.
	Ingest Granule	EcInReqMgr	EcInGran	Distributed Object	Request Manager packages the request into granules and sends them to the Ingest Granule Server.
B.4.3	Create Staging Disk	EcInGran	EcStStagingDiskServer	Distributed Object	Ingest creates Staging Disk, which allocates space and passes back a reference to that disk space. The reference to the Staging Disk is to the Staging Disk Server specified in the Ingest dataabse, based on data type. The amount of staging disk to request is determined from the DAN.
B.4.4	Allocate Media Resource	EcInGran	EcDsStFtpIngestServer	Distributed Object	Ingest now creates the Resource manager for its ftp server via a Resource Manager Factory. Ingest knows that this request is via ftp from a database lookup, keyed on the Data Source (LPS). The correct resource manager is determined from the MediaType handed to the resource factory (IngestFtp, in this case). The correct IngestFtpServer resource is determined from configuration within the resource factory.
B.4.5	Ftp Get files	EcInGran	EcDsStFtpIngestServer	Distributed Object	Ingest directs the ftpserver to get the files from the host and location, as indicated in the DAN, placing them on the staging disk.
B.4.6	ftp Files	EcDsStFtpIngestServer	CSS	rpc	Ftp get files from location.

Table 3.4.5.3-1. Component Interaction Table: Landsat Scenario, LPS Data Insertion (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.5.1	Connect to SDSRV	EcInGran	EcDsScience DataServer	Distributed Object	Ingest begins a session with the SDSRV by connecting. The correct SDSRV is determined during EcInReqMgr startup, from Advertising, based on the data type.
B.5.2	Get Metadata Configuration File	EcInGran	EcDsScience DataServer	Distributed Object	Ingest requests the metadata configuration file (mcf) for the data being inserted. The data types being inserted are derived from the DAN messages sent by LPS.
B.5.3	Validate Metadata	EcInGran	EcDsScience DataServer	Distributed Object	After building the granule's metadata file, Ingest asks SDSRV to validate the metadata, based on the granule's data type.
B.5.4	Insert Data	EcInGran	EcDsScience DataServer	Distributed Object	Ingest requests that the received files for the LPS L0R Subinterval are inserted into the Data Server. An Insert request, containing the names of the files comprising the subinterval, is created. The structure of the Insert Request is hardcoded in the granule server. SDSRV re-validates metadata and determines the archived names of the files.
B.5.5	STMGT Store	EcDsScienceDataServer	EcDsStArchiveServer	Distributed Object	SDSRV requests that the files are archived. The archive server reads the inserted files directly from the Ingest staging disks that they are residing on. The correct archive object to request is determined from collection level metadata for the ESDT, defined in the ESDT's descriptor.
B.5.6	Adding a Granule to Inventory	EcDsScienceDataServer	Sybase/SQS	CtLib	The validated metadata is parsed and added to the inventory of the SDSRV.
B.6.1	Trigger Event	EcDsScienceDataServer	EcSbSubServer	Distributed Object	Upon successful insertion of L0R subinterval granule, the L7OR:Insert event is triggered. The correct subscription server is determined from SDSRV configuration. The correct event to trigger are determined from the events file which was populated during ESDT installation. Provided with the event triggering is the UR of the inserted granule.
B.6.2	Fire Subscriptions	EcSbSubServer	Sybase	CtLib	SBSRV queries the Sybase database determining which subscriptions need to be activated, or fired. Each query "hit" is an activated subscription and will execute independently.
B.7.1	Derive Scenes	EcInGran			The Ingest granule server component is hardcoded to derive and insert scene data whenever a LPS subinterval is inserted.
B.7.2	Get Metadata Configuration File	EcInGran	EcDsScience DataServer	Distributed Object	Ingest requests the metadata configuration file (mcf) for the scene data being inserted.

Table 3.4.5.3-1. Component Interaction Table: Landsat Scenario, LPS Data Insertion (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.7.3	Validate Metadata	EcInGran	EcDsScience DataServer	Distributed Object	After building a metadata file for each virtual scene, Ingest asks SDSRV to validate the metadata, based on the granule's data type.
B.7.4	Insert Data	EcInGran	EcDsScience DataServer	Distributed Object	Ingest requests that the virtual scene granules are inserted into the Data Server. An Insert request, containing the name of the metadata file of the scene, is created. The structure of the Insert Request is hardcoded within the granule server process. SDSRV re-validates metadata file.
B.7.5	Adding a Granule to Inventory	EcDsScienceDataServer	Sybase/SQS	CtLib	The validated metadata is parsed and added to the inventory of the SDSRV.
B.8.1	Trigger Event	EcDsScienceDataServer	EcSbSubServer	Distributed Object	Upon successful insertion of each Landsat scene virtual granule, the L7WRS:Insert event is triggered. This is a qualified event. The scene's spatial metadata is passed along with the trigger. The correct subscription server is determined from SDSRV configuration. The correct events to trigger are determined from the events file which was populated when the ESDT was installed. Provided with the event triggering is the UR of the inserted granule.
B.8.2	Fire Subscriptions	EcSbSubServer	Sybase	CtLib	SBSRV queries the Sybase database determining which subscriptions need to be activated, or fired. Each query "hit" is an activated subscription and will execute independently.
B.9.1	Get Metadata Configuration File	EcInGran	EcDsScience DataServer	Distributed Object	Based on the browse file indicator in the DAN Ingest begins the insertion of browse data. Ingest requests the metadata configuration file (mcf) for the browse data being inserted.
B.9.2	Insert Data	EcInGran	EcDsScience DataServer	Distributed Object	After deriving browse data and metadata from the subinterval, Ingest requests that the browse files for the LPS L0R scenes are inserted into the Data Server. An Insert request, containing the names of the files comprising the browse, is created. The structure of the Insert Request is hardcoded in the granule server. SDSRV re-validates metadata and determines the archived names of the files.
B.9.3	STMGT Store	EcDsScienceDataServer	EcDsStArchiveServer	Distributed Object	SDSRV requests that the files are archived. The archive server reads the inserted files directly from the Ingest staging disks that they are residing on. The correct archive object to request is determined from collection level metadata for the ESDT, defined in the ESDT's descriptor.

Table 3.4.5.3-1. Component Interaction Table: Landsat Scenario, LPS Data Insertion (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.9.4	Adding a Granule to Inventory	EcDsScienceDataServer	Sybase/SQS	CtLib	The validated metadata is parsed and added to the inventory of the SDSRV.
B.9.5	Completion Callback	EcInReqMgr	IcInGran	rpc	The Ingest Granule Server sends a completion callback to the Ingest Request Manager when the processing of the granule is complete.
B.10.1	Data Delivery status	EcInReqMgr	EcCsLandsat7Gateway	rpc	Ingest creates a Data Delivery Notice indicating successful insertion of LPS data.
B.11.1	Data Delivery Notice	EcCsLandsat7Gateway	LPS	rpc	L7 Gateway sends the Data Delivery Notice to LPS. LPS responds with a Data Delivery Acknowledgement (DDA).

3.4.6 Standing Order Support Thread

3.4.6.1 Interaction Diagram - Domain View

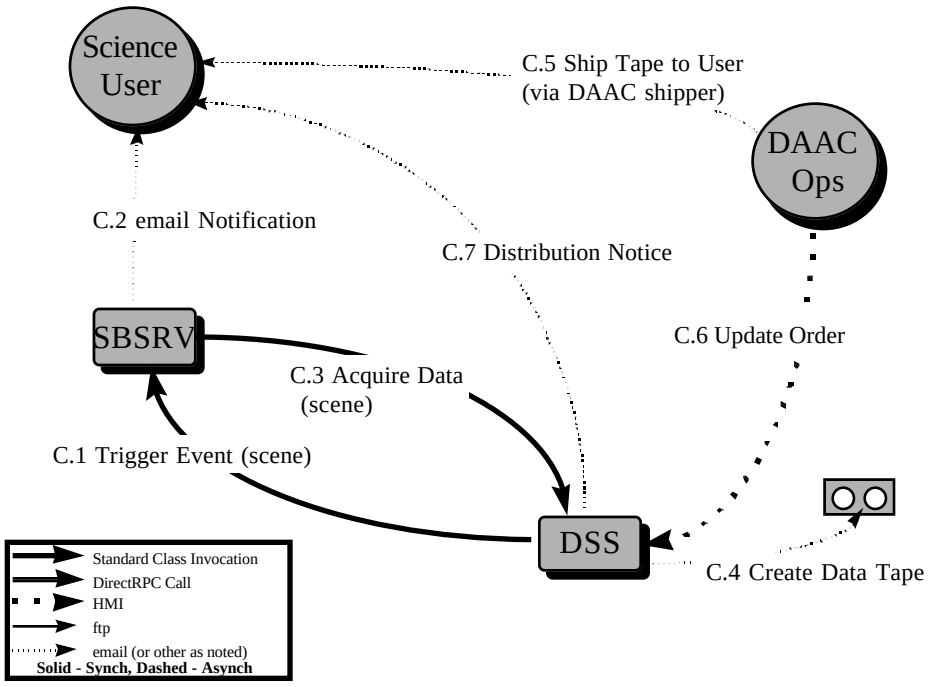

Figure 3.4.6.1-1. Standing Order Support Interaction Diagram

3.4.6.2 Interaction Table - Domain View

Table 3.4.6.2-1. Interaction Table - Domain View: Landsat 7 Scenario, Standing Order Support

Step	Interaction	Interface Client	Interface Provider	Data Issues	Preconditions	Description
C.1	Trigger Event	SDSRV	SBSRV			This thread picks up with fulfilling the standing order WRS scene data. This is a qualified event with the spatial attributes of the scene specified. In this scenario, the newly inserted scene is within the spatial constraints specified by the Science User. This is the same step as B.7 of this Landsat 7 scenario.
C.2	Notification	SBSRV	Science User			Send email notification to Science User, notifying that there is a newly inserted WRS scene granule that meets the specified spatial area of interest. Notification message includes the UR of the scene granule as well as the qualification criteria that it matched.
C.3	Acquire Data	SBSRV	SDSRV			SBSRV fulfills the standing order by the Science User, for WRS scene granules. Request to Acquire data, via 8mm tape, for the Science User.
C.4	Create Data Tape	DSS	Tape device	1 WRS scene @812MB (nominally)		Data Server copies the WRS Scene granule's files to 8mm tape and marks the order as "Ready to Ship".
C.5	Ship Tape to User	DAAC Ingest/Distribution Technician	Science User			DAAC Ingest/Distribution Technician collects tape, media label shipping label and packing list. They label tape, enclose tape and packing list in shipping container and label shipping container. DAAC uses commercial shipping vendor for delivery to Science User.
C.6	Update Order	DAAC Ingest/Distribution Technician	DSS			DAAC Ingest/Distribution Technician marks order as "Shipped".
C.7	Distribution Notice	DSS	Science User			Send email notification to Science User, notifying that the newly inserted WRS scene of interest has been shipped to their shipping address.

3.4.6.3 Component Interaction Table

Table 3.4.6.3-1. Component Interaction Table: Landsat Scenario, Standing Order Support

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
C.1.1	Trigger Event	EcDsScienceDataServer	EcSbSubServer	Distributed Object	(Same as B.7.1) Upon successful insertion of L7 WRS scene granules, the L7WRS:Insert event is triggered, for each granule. These are qualified events. Along with the trigger are spatial metadata qualifiers. The correct subscription server is determined from SDSRV configuration The correct event to trigger is determined from the events file, which was populated during ESDT installation. Provided with the event triggering is the UR of the inserted granule.
C.1.2	Fire Subscriptions	EcSbSubServer	Sybase	CtLib	SBSRV queries the Sybase database determining which subscriptions need to be activated, or fired. Each query "hit" is an activated subscription and will execute independently. Currently all the subscriptions on this qualified event are not qualified, so all subscriptions are "hits".
C.2.1	Send E-mail	EcSbSubServer	Science User		The SBSRV builds an email notification that the user's subscription on the WRS scenes insert event has been fired. This notification will identify the Event, the subscription ID, the UR that was inserted and the previously supplied UserString.
C.3.1	Connect to SDSRV	EcSbSubServer	EcDsScienceDataServer	Distributed Object	In order to fulfill a standing order for the L7 WRS Scene data, the SBSRV begins a session with the SDSRV, on behalf of the subscription user. The correct SDSRV is determined from the Granule UR provided with the event triggering.
C.3.2	Add granules to Session	EcSbSubServer	EcDsScienceDataServer	Distributed Object	The SBSRV establishes the data context of the session with the SDSRV by adding the input granules to the session. The GranuleUR of each input granule is added to the ESDTReferenceCollection.
C.3.3	Retrieve Granule Metadata from Inventory	EcDsScienceDataServer	Sybase/SQS	CtLib	SDSRV completes establishing the data context by retrieving the metadata for the requested granules from the Sybase/SQS database. The metadata for each granule is passed back to the reference objects for each granule.

Table 3.4.6.3-1. Component Interaction Table: Landsat Scenario, Standing Order Support (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
C.3.4	Acquire Data	EcSbSubServer	EcDsScienceDataServer	Distributed Object	SBSRV fulfills the standing order for the L7 WRS Scene granule by submitting an Acquire request for the granule. The Acquire request is for an 8mm tape of all granules in the ESDTReferenceCollection. This request is asynchronous, meaning that the return of the submit call of the request will only contain the status of the request's submittal. The request asks for an email notification to be email'd to the user. The Acquire request structure is hardcoded within the Subscription Server. For the granule referenced in the Acquire request, the SDSRV creates a file containing the granules metadata before passing to Distribution.
C.3.5	Create Staging Disk	EcDsScienceDataServer	EcStStagingDiskServer	Distributed Object	SDSRV creates Staging Disk for working space, scene files and metadata files, which allocates space and passes back a reference to that disk space. The reference to the Staging Disk is determined from the ArchiveID metadata of the granule to be distributed The amount of staging disk to request is determined from collection level metadata from the ESDT's Descriptor file.
C.3.6	STMGT Retrieve	EcDsScienceDataServer	EcDsStArchiveServer	Distributed Object	DDIST requests that STMGT retrieve the subinterval granule files that are to be subsetted. For a WRS Scene, these files include Band files. This will result in the files being staged to the working staging disk area. The correct archive object to request is determined from the information provided by the SDSRV in the distribution request.
C.3.7	L7 Scene creation	EcDsScienceData Server			Since L7 WRS Scenes are virtual granules (i.e. the granules are represented in the inventory with their metadata, but the files which contain the data don't actually exist), the scene files must be extracted from the scene's parent subinterval. The scenes files are created using internal subinterval subsetting methods. The subsetted files include Band files, MTA, PCD Calibration files, and Mirror Scan correction data files. A Metadata file is created for the new WRS Scene.
C.3.8	Distribute Granules, Synchronous	EcDsScienceDataServer	EcDsDdRequestMgrMain	Distributed Object	SDSRV submits a request to Data Distribution. The request includes, for the WRS Scene granule, a reference to the subsetted and metadata files as well as the data files that are still in the archive. Other parameters from the Acquire request are passed to DDIST.

Table 3.4.6.3-1. Component Interaction Table: Landsat Scenario, Standing Order Support (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
C.3.9	Create Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST creates Staging Disk for the granule files in the archive. This allocates space and passes back a reference to that disk space. The reference to the Staging Disk is determined from the information passed by the SDSRV in the distribution request ,which was the ArchiveID metadata parameter of the granule to be staged. The amount of staging disk to request is calculated from the file sizes in the information passed in the Distribution Request.
C.3.10	STMGT Retrieve	EcDsDdRequestMgrMain	EcDsStArchiveServer	Distributed Object	DDIST requests that STMGT retrieve the granule files that are still archived. For a WRS Scene, these files include MSCD, Calibration File, MTA, PCD, and CPF files. This will result in the file being staged to read-only cache disks. This means that all files needed to fulfill the distribution request are on disk, and ready to be copied. The correct archive object to request is determined from the information provided by the SDSRV in the distribution request. This returns references to the files in the read-only cache.
C.3.11	Link files to Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST links the files from the read-only cache into the staging disk.
C.3.12	Copy files to Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST copies the subsetted scene and metadata files from the SDSRV's Staging Disk into the staging disk.
C.4.1	Allocate Media Resource	EcDsDdRequestMgrMain	EcDsSt8mmTapeServer	Distributed Object	DDIST now creates the Resource manager for 8mm via a Resource Manager Factory. The correct resource manager is determined from the MediaType handed to the resource factory (8mm, in this case). The correct 8mm resource is determined from configuration within the resource factory.
C.4.2	Write files to 8mm tape	EcDsDdRequestMgrMain	EcDsSt8mmTapeServer	Distributed Object	DDIST requests that the tapes held in staging disk be copied to the 8mm device. Upon completion of the files being copied, the state of the distribution request is marked as "Ready for Shipment", which is displayed on the operator GUI. A packing list is generated.
C.5.1	Determine request is ready for shipping	DAAC Ops - Distribution Technician	EcDsDdistGui	GUI	Operator notes that the request is ready for shipping and that it includes the 8mm tape just produced. The 8mm tape slot and stacker id are included on the request screen, so that the operator knows which tapes to ship.

Table 3.4.6.3-1. Component Interaction Table: Landsat Scenario, Standing Order Support (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
C.5.2	Ship Tapes	DAAC Ops - Data Technician			The labels for all media., as well as a shipping label for the package, are created manually. Using commercial shipping vendors (determined by DAAC policy), the DAAC Data Technician labels the tape, packages the tape(s) and packing list, labels the package and ships to address provided with the request.
C.6.1	Mark as Shipped	DAAC Ops - Data Technician	EcDsDdist Gui	GUI	Using the DSS GUI, the Data Technician marks the request as “Shipped”.
C.6.2	Update Distribution Request	EcDsDdist Gui	EcDsDdRequestMgrMain	Distributed Object	DDIST updates the state of the request to “Shipped”.
C7.1	Build Distribution Notice	EcDsDdRequestMgrMain			The DDIST builds an email notification that the user’s order has been fulfilled. This notification will include the media id, type and format, as well as UR, type and file names and sizes for each granule.
C7.2	Send E-mail	EcDsDdRequestMgrMain	Science User	email	DDIST sends the distribution notice to the user via email. Since Standing Orders are not currently tracked orders, and the user DDIST sends the notification to is determined from the Order, the notice is currently sent to a pre-configured default Email address, for DAAC Distribution Technician parsing and forwarding.

3.4.7 IAS Data Insertion Thread

3.4.7.1 Interaction Diagram - Domain View

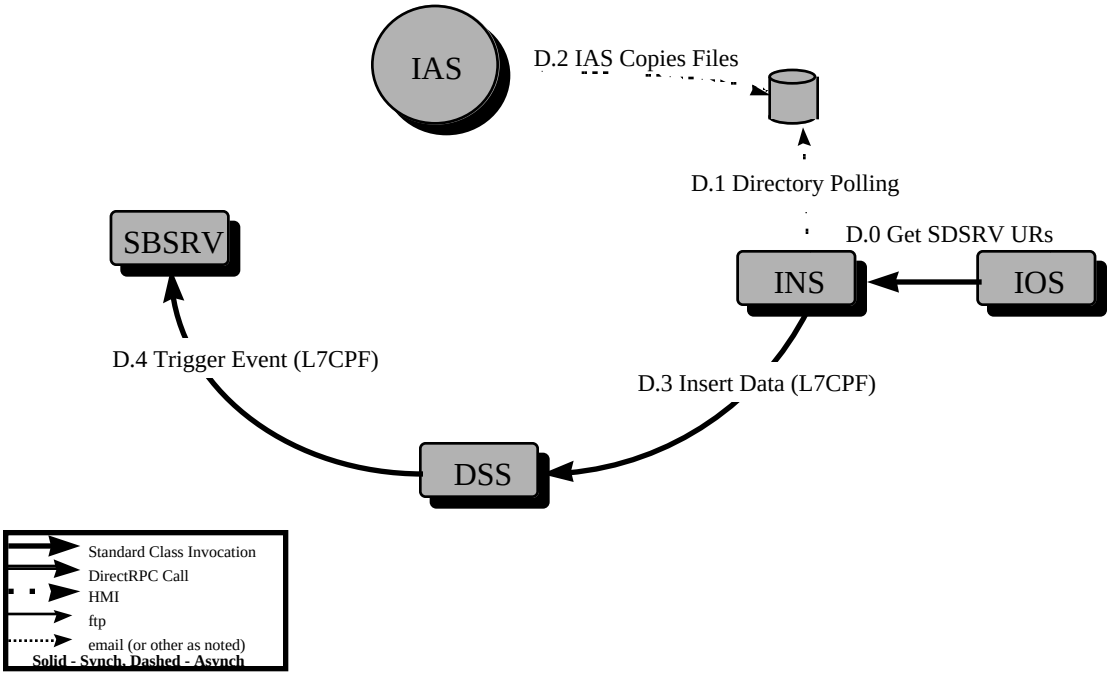

Figure 3.4.67.1-1. IAS Data Insertion Interaction Diagram

3.4.7.2 Interaction Table - Domain View

Table 3.4.7.2-1. Interaction Table - Domain View: Landsat 7 Scenario, IAS Data Insertion

Step	Interaction	Interface Client	Interface Provider	Data Issues	Preconditions	Description
D.0	Get SDSRV URs	INS	IOS			Upon startup, Ingest gets the SDSRV URs for each data type in its database.
D.1	Polling	INS	directory		Entire step is really a precondition.	When system is started, Ingest begins polling a directory, looking for files that meet the following standard: *.PDR, in the preconfigured directory.
D.2	Copy Files	IAS	directory		IAS knows the host and directory to place files.	IAS copies the CPF, PDR and metadata files to the directory which Ingest is polling.
D.3	Insert Data	INS	DSS	1 L7CPF @ 127MB	L7CPF ESDT	Ingest inserts the new CPF granule into the Data Server.
D.4	Trigger Event	DSS	SBSRV			Upon successful completion of insertion of L7CPF, the L7CPF:Insert event is triggered.

3.4.7.3 Component Interaction Table

Table 3.4.7.3-1. Component Interaction Table: Landsat Scenario, IAS Data Insertion (1 of 2)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
D.0.0	Get SDSRV URs from IOS	EcInReqMgr	EcIoAdServer	Distributed Object	Upon startup, Ingest Request Manager requests the SDSRV URs fro each data type in its database.
D.1.1	Ingest Polling	EcInPolling			Ingest begins polling the configured directory. It will periodically look for files named *.PDR. The polling periodicity is determined from a configuration file. The mask of the file to look for is determined from configuration.
D.2.1	IAS Copies Files	IAS		ftp	IAS ftp's the Calibration Parameter File to the predetermined directory. Location, directory, username and password are as per the L7-ECS ICD.
D.3.1	Polling Detects Files	EcInPolling			Ingest Polling detects files matching the *.PDR mask.
D.3.2	Ingest Request	EcInPolling	EcInReqMgr	Distributed Object	Polling Ingest process packages the Data Transfer messages into the appropriate Ingest Requests. The data source (IAS), defined on startup, is passed to the Ingest Request Manager.
	Ingest Granule	EcInReqMgr	EcInGran	Distributed Object	Request Manager packages the request into granules and sends them to the Ingest Granule Server.
D.3.3	Create Staging Disk	EcInGran	EcStStagingDiskServer	Distributed Object	Ingest creates Staging Disk. The reference to the Staging Disk server is determined from the Ingest Database. The amount of staging disk to request is determined from the *.PDR file.
D.3.4	Allocate Media Resource	EcInGran	EcDsStFtpIngestServer	Distributed Object	Ingest now creates the Resource manager for its ftp server via a Resource Manager Factory. Ingest knows that this request is via ftp from a database lookup, keyed on the Data Source (IAS). The correct resource manager is determined from the MediaType handed to the resource factory (IngestFtp, in this case). The correct IngestFtpServer resource is determined from configuration within the resource factory.
D.3.5	Ftp Get files	EcInGran	EcDsStFtpIngestServer	Distributed Object	Ingest directs the ftpserver to get the files from the host and location, as indicated in the *.PDR file, placing them on the staging disk.
D.3.6	ftp Files	EcDsStFtpIngestServer	CSS	rpc	Ftp get files from location.
D.3.7	Connect to SDSRV	EcInGran	EcDsScienceDataServer	Distributed Object	Ingest begins a session with the SDSRV by connecting. The correct SDSRV is determined during EcInReqMgr startup, from Advertising, based on the data type.

Table 3.4.7.3-1. Component Interaction Table: Landsat Scenario, IAS Data Insertion (2 of 2)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
D.3.8	Get Metadata Configuration File	EcInGran	EcDsScience DataServer	Distributed Object	Ingest requests the metadata configuration file (mcf) for the data being inserted. The data types being inserted are derived from the *.PDR file sent by IAS.
D.3.9	Validate Metadata	EcInGran	EcDsScience DataServer	Distributed Object	After building a metadata file for the CPF granule, Ingest asks SDSRV to validate the metadata, based on the granule's data type.
D.3.10	Insert Data	EcInGran	EcDsScience DataServer	Distributed Object	Ingest requests that the received files for the L7CPF are inserted into the Data Server. An Insert request, containing the names of the files comprising the CPF granule, is created. The structure of the Insert Request is hardcoded in the granule server process. SDSRV re-validates metadata and determines the archived names of the files.
D.3.11	STMGT Store	EcDsScience DataServer	EcDsStArchiveServer	Distributed Object	SDSRV requests that the CPF is archived. The archive server reads the inserted files directly from the Ingest staging disks that they are residing on. The correct archive object to request is determined from collection level metadata for the ESDT, defined in the ESDT's descriptor.
D.3.12	Adding a Granule to Inventory	EcDsScience DataServer	Sybase/SQS	CtLib	The validated metadata is parsed and added to the inventory of the SDSRV.
B.3.13	Completion Callback	EcInReqMgr	EcInGran	rpc	The Ingest Granule Server sends a completion callback to the Ingest.
D.3.14	IAS Data Response	EcInReqMgr	IAS	file	Ingest places the completion status in a Product Delivery Notice (PDN) file, which is placed in a directoy accessible to IAS, as per the ECS-Landsat ICD.
D.4.1	Trigger Event	EcDsScience DataServer	EcSbSubServer	Distributed Object	Upon successful insertion of L7 CPF granule, the L7CPF:Insert event is triggered. The correct subscription server is determined from SDSRV configuration. The correct events to trigger are determined from the events file, which was populated during ESDT installation. Provided with the event triggering is the UR of the inserted granule.
D.4.2	Fire Subscriptions	EcSbSubServer	Sybase	CtLib	SBSRV queries the Sybase database determining which subscriptions need to be activated, or fired. Each query "hit" is an activated subscription and will execute independently.

3.4.8 Search and Browse Thread

3.4.8.1 Interaction Diagram - Domain View

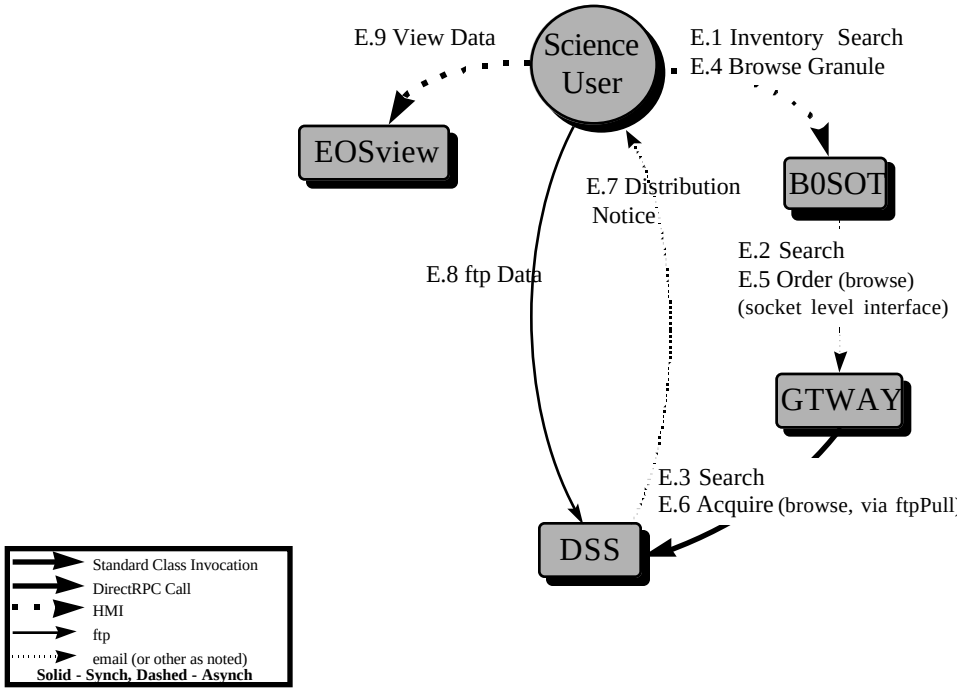

Figure 3.4.8.1-1. Search and Browse Interation Diagram

3.4.8.2 Interaction Table - Domain View

Table 3.4.8.2-1. Interaction Table - Domain View: Landsat 7 Scenario, Search and Browse

Step	Interaction	Interface Client	Interface Provider	Data Issues	Preconditions	Description
E.1	Inventory Search	Science User	B0SOT			Upon notification that there are new scenes available, the Science User decides to look for additional scenes of interest. First, the user invokes B0SOT and searches for scenes over another area of interest.
E.2	Search	B0SOT	GTWAY			B0SOT submits the Science User's search criteria to the V0 Gateway in ODL format, via a well defined socket.
E.3	Search	GTWAY	SDSRV			The V0 gateway translates the Search criteria from ODL to a query object (using GlParameters), and submits that query to the Search service. The results of this Search are returned synchronously, and are passed back to B0SOT, which displays them to the DAAC Operator.
E.4	Browse Granule	Science User	B0SOT			User decides some of these granules might be of interest, so before ordering them decides to get a browse image of one to verify.
E.5	Order	B0SOT	V0GWY			B0SOT submits ftp Browse Request to the V0 Gateway in ODL format via a well defined socket-level interface.
E.6	Acquire Data	V0GWY	SDSRV			V0GWY submits an Acquire request for the browse granule, via ftpPull.
E.7	Distribution Notice	DSS	Science User			Send email notification to Science User, notifying that the browse granule is now available for ftp access.
E.8	ftp Data	Science User	DSS	1 browse granule @ 1MB		Scientist ftp's browse granule to their workstation.
E.9	View Data	Science User	EOSView			Science User invokes EOSView, and views the selected scene's browse image.

3.4.8.3 Component Interaction Table

Table 3.4.8.3-1. Component Interaction Table: Landsat Scenario, Search and Browse

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
E.1.1	Startup B0SOT	Science User	xims	Xevent	Science User invokes the B0SOT GUI Application. The Science User has already been logged on the Science Desktop and begins B0SOT by double-clicking on an icon.
E.1.2	Select Inventory Search, Provide Query constraints, Submit Query	Science User	xims	GUI	The Science User provides search constraints and the products desired. When query constraints are completed, the query is submitted.
E.2.1	V0 Gateway Inventory	xims	EcDmV0ToEcsGateway	ODL, over sockets	B0SOT submits a search to the V0 Gateway, by converting the search criteria into an ODL structure and passing that structure to a socket provided by the Gateway. The correct socket is determined from configuration information in the Valids file.
E.3.1	Establish ECS User	EcDmV0ToEcsGateway	MsAcManager	Distributed Object	V0 Gateway retrieves the User Profile using ECS Authenticator from ODL message, which includes an encrypted UserID and Password. The User Registration Server is replicated across DAACs, so the connection is made to the local User Registration Server.
E.3.2	Translate Query	EcDmV0ToEcsGateway	EcDmDictServer	Ctlib (RWDbTool)	Gateway translates the V0 terms from ODL into ECS names for quirey submittal. Interface currently is directly to Data Dictionary database. Database name is retrieved from configuration file.
E.3.3	Connect to SDSRV	EcDmV0ToEcsGateway	EcDsScienceDataServer	Distributed Object	The Gateway first connects to the SDSRV. The correct SDSRV is determined by configuration file.
E.3.4	SDSRV Query	EcDmV0ToEcsGateway	EcDsScienceDataServer	Distributed Object	The Gateway translates the query into a DsClQuery object. This object is handed to the Search interface of the DsClESDTReferenceCollector. This Search method is synchronous, so the results of the search will be returned to the calling function.
E.3.5	Querying the Inventory	EcDsScienceDataServer	Sybase/SQS	CtLib	The SDSRV breaks down the Query object and translates it into a sequence of calls to the inventory database. Resultant rows are converted into data granules, each with their metadata extracted from the database. These results are packaged and returned to the Query client.

Table 3.4.8.3-1. Component Interaction Table: Landsat Scenario, Search and Browse (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
E.3.6	Result Retrieval	xims	EcDmV0ToEcsGateway	ODL, over Sockets	When the Gateway gets the results, they are translated into ODL, and passed back to the B0SOT tool. The correct socket for sending results to B0SOT is the one used to submit the query. B0SOT then displays the results of the query to the user.
E.4.1	B0SOT ftp Browse	Science User	xims	GUI	Science User decides to browse a specified granule. They click the Browse field, then “Go To Browse”.
E.5.1	V0 ftp Browse	xims	EcDmV0ToEcsGateway	ODL, over sockets	B0SOT submits an Ftp Pull Request for the browse to the V0 Gateway and sending it the Order structure to a socket provided by the Gateway. The correct socket is determined from configuration file.
E.5.2	Establish ECS User	EcDmV0ToEcsGateway	MsAcManager	Distributed Object	V0 Gateway retrieves the User Profile using ECS Authenticator from ODL message, which includes an encrypted UserID and Password. The User Registration Server is replicated across DAACs, so the connection is made to the local User Registration Server.
E.5.3	Translate Terms	EcDmV0ToEcsGateway	EcDmDictServer	Ctlib (RWDbTool)	Gateway translates the V0 terms from ODL into ECS names for request submittal. Interface currently is directly to Data Dictionary database. Database name is retrieved from configuration file.
E.5.4	Connect to SDSRV	EcDmV0ToEcsGateway	EcDsScienceDataServer	Distributed Object	The V0Gateway begins a session with the SDSRV, on behalf of the science user. The correct SDSRV is determined by the UR of the granule whose browse is being requested.
E.5.5	Add granules to Session	EcDmV0ToEcsGateway	EcDsScienceDataServer	Distributed Object	The V0 Gateway establishes the data context of the session with the SDSRV by adding granules to the session. The GranuleUR of the granule to be browsed is added to the ESDTReferenceCollection.
E.5.6	Retrieve Granule Metadata from Inventory	EcDsScienceDataServer	Sybase/SQS	CtLib	SDSRV completes establishing the data context by retrieving the metadata for the requested granules from the Sybase/SQS database. The metadata for each granule is passed back to the reference objects for each granule.
E.5.7	Inspect Granule Value Parameters	EcDmV0ToEcsGateway	EcDsScienceDataServer	Distributed Object	The V0 Gateway determines the UR of the Browse granule to acquire by inspecting the “BrowseId” metadata attribute of the granule to be browsed.
E.5.8	Add granules to Session	EcDmV0ToEcsGateway	EcDsScienceDataServer	Distributed Object	The V0 Gateway adds the Browse granule to the data context of the session with the SDSRV by adding its UR to the session.

Table 3.4.8.3-1. Component Interaction Table: Landsat Scenario, Search and Browse (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
E.5.9	Retrieve Granule Metadata from Inventory	EcDsScienceDataServer	Sybase/SQS	CtLib	SDSRV completes adding the browse granule to the session by retrieving the metadata for the requested granules from the Sybase/SQS database. The metadata for each granule is passed back to the reference objects for each granule.
E.6.1	Acquire Data	EcDmV0ToEcsGateway	EcDsScienceDataServer	Distributed Object	The V0 Gateway submits an Acquire request for the granule. The Acquire request is for an ftpPull of the browse granule in the ESDTReferenceCollection. This request is asynchronous, meaning that the return of the submit call of the request will only contain the status of the request's submittal. This request asks for a distribution notice to be email'd. The Acquire request structure is hard coded using the consts in the DDIST provided header file.
E.6.2	Create Staging Disk	EcDsScienceDataServer	EcStStagingDiskServer	Distributed Object	SDSRV creates Staging Disk for the browse granule's metadata file, which allocates space and passes back a reference to that disk space. The reference to the Staging Disk is determined from the ArchiveID metadata of the browse granule to be distributed The amount of staging disk to request is determined from collection level metadata from the Browse Descriptor file.
E.6.3	Create Metadata file	EcDsScienceDataServer			For the browse granule referenced in the Acquire request, the SDSRV creates a file containing the granule's metadata before passing to Distribution.
E.6.4	Distribute Granules, Synchronous	EcDsScienceDataServer	EcDsDdRequestMgrMain	Distributed Object	SDSRV submits a request to Data Distribution. The request includes, for the granule, a reference to the metadata file as well as the data file. Other parameters from the Acquire request are passed to DDIST.
E.6.5	Create Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST creates Staging Disk for the granule files in the archive. This allocates space and passes back a reference to that disk space. The reference to the Staging Disk is determined from the information passed by the SDSRV in the distribution request ,which was the ArchiveID metadata parameter of the granule to be staged. The amount of staging disk to request is calculated from the file sizes in the information passed in the Distribution Request.
E.6.6	STMGT Retrieve	EcDsDdRequestMgrMain	EcDsStArchiveServer	Distributed Object	DDIST requests that STMGT retrieve the browse granule file that is archived. This will result in the file being staged to read-only cache disks. This means that all files needed to fulfill the distribution request are on disk, and ready to be copied. The correct archive object to request is determined from the information provided by the SDSRV in the distribution request. This returns references to the files in the read-only cache.

Table 3.4.8.3-1. Component Interaction Table: Landsat Scenario, Search and Browse (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
E.6.7	Link files to Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST links the browse file from the read-only cache into the staging disk.
E.6.8	Copy files to Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST copies the browse metadata file from the SDSRV's Staging Disk into the staging disk.
E.6.9	DDIST Pull No Tar	EcDsDdRequestMgrMain	EcDsStFtpDisServer	Distributed Object	DDIST now creates the Resource manager for ftpPull via a Resource Manager Factory. The correct resource manager is determined from the MediaType handed to the resource factory (ftpPull, in this case). The correct FtpServer is determined from configuration within the resource factory.
E.7.1	Build Distribution Notice	EcDsDdRequestMgrMain			DDIST builds a distribution notice indicating that the browse data is now available. The notice includes the UR of the browse granule, name of the browse file, the host and directory name where it is available and how long it will be available.
E.7.2	Send E-mail	EcDsDdRequestMgrMain	Science User		The distribution notice is email'd to the Science User.
E.8.1	User Ftp's Data	Scientist's ftp utility	ftp_popen	ftp	The scientist uses ftp to get the browse file.
E.9.1	Invoke EOSView	Science User	EOSView	Xevent	Science User begins the EOSView application. While logged on to the Science Desktop, the user double clicks the EOSView icon.
E.9.2	Display Browse File	Science User	EOSView	GUI	The Science User specifies which file to display and sets visualization parameters. The browse file is now displayed for the user.

3.4.9 Ordering WRS Scenes Thread

3.4.9.1 Interaction Diagram - Domain View

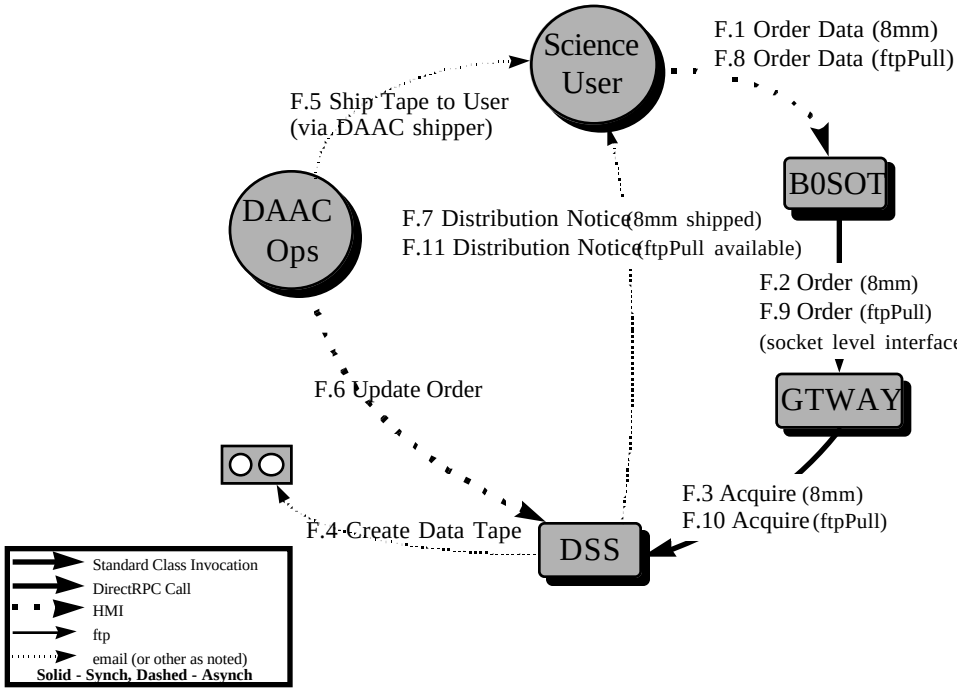

Figure 3.4.9.1-1. Ordering WRS Scenes Interaction Diagram

3.4.9.2 Interaction Table - Domain View

Table 3.4.9.2-1. Interaction Table - Domain View: Landsat 7 Scenario, Ordering WRS Scenes

Step	Interaction	Interface Client	Interface Provider	Data Issues	Preconditions	Description
F.1	Order Data	Science User	B0SOT			Scientist decides to order a number of L7 WRS scenes. Using B0SOT, cientist selects the scenes of interest and selects order via 8mm tape.
F.2	Order	B0SOT	GTWAY			B0SOT submits the Science User's order to the V0 Gateway in ODL format, via a well defined socket.
F.3	Acquire	GTWAY	SDSRV	1 L7WRS granules @ 812 MB		The V0 gateway translates the order into an Acquire request. The request is for a set of WRS scenes, via 8mm tape. To fulfill the acquire of WRS scenes, which are virtual granules, the Data Server derives the scenes from their parent subinterval, using internally available subsetting services.
F.4	Create Tape	DSS	Tape Device			Data Server copies the WRS Scenes granule's files to 8mm tape and marks the order as "Ready to Ship".
F.5	Ship Tape to User	DAAC Ingest Distribution Technician	Science User			DAAC Ingest/Distribution Technician collects tape, media label shipping label and packing list. They label tape, enclose tape and packing list in shipping container and label shipping container. DAAC uses commercial shipping vendor for delivery to Science User.
F.6	Update Order	DAAC Ops (Distribution Technician)	DSS			Data Technician marks order as "Shipped".
F.7	Distribution Notice	DSS	Science User			Send email notification to Science User, notifying that the ordered WRS scenes have been shipped to their shipping address.
F.8	Order Data	Science User	B0SOT			Scientist decides to order a particular scene of high interest, via ftpPull.
F.9	Order	B0SOT	V0GWY			B0SOT submits the Science User's order to the V0 Gateway in ODL format, via a well defined socket.
F.10	Acquire	V0GWY	DSS	1 WRS scene @812MB (nominally)		The V0 gateway translates the order into an Acquire request. The request is for a set of WRS scenes, via ftpPull. To fulfill the acquire of WRS scenes, which are virtual granules, the Data Server derives the scenes from their parent subinterval, using internally available subsetting services.
F.11	Distribution Notice	DSS	Science User			Send email notification to Science User, notifying them that the requested scene is now available for ftp access.

3.4.9.3 Component Interaction Table

Table 3.4.9.3-1. Component Interaction Table: Landsat Scenario, Ordering WRS Scenes

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
F.1.1	Select data to be ordered via 8mm tape	Science User	xims	GUI	The Science User selects a set of WRS scenes to order for delivery via 8mm tape. When the order is complete it is submitted to the V0 Gateway.
F.2.1	V0 Gateway Order	xims	EcDmV0ToEcsGateway	ODL, over sockets	B0SOT submits an order to the V0 Gateway, by converting the order into an ODL structure and passing that structure to a socket provided by the Gateway. The correct socket is determined from configuration information in the Valids file.
F.2.2	Establish ECS User	EcDmV0ToEcsGateway	MsAcManager	Distributed Object	V0 Gateway retrieves the User Profile using ECS Authenticator from ODL message, which is an encrypted UserID and Password. The User Registration Server is replicated across DAACs, so the connection is made to the local User Registration Server.
F.2.3	Translate Terms	EcDmV0ToEcsGateway	EcDmDictServer	Ctlib (RWDbTool)	Gateway translates the V0 terms from ODL into ECS names for request submittal. Interface currently is directly to Data Dictionary database. Database name is retrieved from configuration file.
F.2.4	Create Tracked Order	EcDmV0ToEcsGateway	MsAcOrderManager	Distributed Object	Create an order to be tracked within ECS. Initial status set to “Pending”.
F.2.5	Persist Tracked Order	MsAcOrderManager	Sybase	Ctlib	Save the tracked order to the order database.
F.2.6	Create Tracked Request	EcDmV0ToEcsGateway	MsAcOrderManager	Distributed Object	Create the Tracked Request within the Order. Initial status set to “Pending”.
F.2.7	Persist Tracked Request	MsAcOrderManager	Sybase	Ctlib	Save the tracked request to the order database.
F.2.8	Connect to SDSRV	EcDmV0ToEcsGateway	EcDsScienceDataServer	Distributed Object	The Gateway first connects to the SDSRV. The correct SDSRV is determined by the UR of a granule that is being ordered.
F.2.9	Add granules to Session	EcDmV0ToEcsGateway	EcDsScienceDataServer	Distributed Object	The V0 Gateway establishes the data context of the session with the SDSRV by adding granules to the session. The GranuleUR of the WRS Scene granules to be ordered are added to the ESDTReferenceCollection.

Table 3.4.9.3-1. Component Interaction Table: Landsat Scenario, Ordering WRS Scenes (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
F.2.10	Retrieve Granule Metadata from Inventory	EcDsScienceDataServer	Sybase/SQS	CtLib	SDSRV completes establishing the data context by retrieving the metadata for the requested granules from the Sybase/SQS database. The metadata for each granule is passed back to the reference objects for each granule.
F.3.1	Acquire Data	EcDmV0ToEcsGateway	EcDsScienceDataServer	Distributed Object	V0Gateway submits the order for WRS Scene granules by submitting an Acquire request for the granules. The Acquire request is for an 8mm tape of all granules in the ESDTReferenceCollection. This request is asynchronous, meaning that the return of the submit call of the request will only contain the status of the request's submittal. The request asks for an email notification to be email'd to the user. The Acquire request structure was determined from the Action submitted with the standing order method.
F.3.2	Create Staging Disk	EcDsScienceDataServer	EcStStagingDiskServer	Distributed Object	SDSRV creates Staging Disk for working space, scene files and metadata files, which allocates space and passes back a reference to that disk space. The reference to the Staging Disk is determined from the ArchiveID metadata of the granule to be distributed The amount of staging disk to request is determined from collection level metadata from the ESDT's Descriptor file.
F.3.3	STMGT Retrieve	EcDsDdRequestMgrMain	EcDsStArchiveServer	Distributed Object	DDIST requests that STMGT retrieve the subinterval granule files that are to be subsetted. For a WRS Scene, these files include Band files, MSCD, Calibration File, MTA, PCD, and CPF files. This will result in the files being staged to the working staging disk area. The correct archive object to request is determined from the information provided by the SDSRV in the distribution request.
F.3.4	L7 Scene creation	EcDsScienceData Server			Since L7 WRS Scenes are virtual granules (i.e. the granules are represented in the inventory with their metadata, but the files which contain the data don't actually exist), the scene files must be extracted from the scene's parent subinterval. The scenes files are created using internal subinterval subsetting methods. The subsetted files include Band files, MSCD, Calibration File, MTA and PCD files. Metadata files for each WRS Scene granule are created.

Table 3.4.9.3-1. Component Interaction Table: Landsat Scenario, Ordering WRS Scenes (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
F.3.5	Distribute Granules, Synchronous	EcDsScienceDataServer	EcDsDdRequestMgrMain	Distributed Object	SDSRV submits a request to Data Distribution. The request includes, for each WRS Scene, a reference to the subsetted and metadata files as well as the data file still in the archive. Other parameters from the Acquire request are passed to DDIST.
F.3.6	Update Tracked Request Status	EcDsDdRequestMgrMain	MsAcOrderManager	Distributed Object	Update the status of the tracked request to “Active”.
F.3.7	Update Tracked Status	EcAcOrderManager	Sybase	Ctlib	Update the Order Tracking Database with new status.
F.3.8	Create Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST creates Staging Disk for the granule files in the archive. This allocates space and passes back a reference to that disk space. The reference to the Staging Disk is determined from the information passed by the SDSRV in the distribution request ,which was the ArchiveID metadata parameter of the granule to be staged. The amount of staging disk to request is calculated from the file sizes in the information passed in the Distribution Request.
F.3.9	Update Tracked Request Status	EcDsDdRequestMgrMain	EcAcOrderManager	Distributed Object	Update the status of the tracked request to “Staging”.
F.3.10	Update Tracked Status	EcAcOrderManager	Sybase	Ctlib	Update the Order Tracking Database with new status.
F.3.11	STMGT Retrieve	EcDsDdRequestMgrMain	EcDsStArchiveServer	Distributed Object	DDIST requests that STMGT retrieve the granule files that are archived. This will result in the file being staged to read-only cache disks. This means that all files needed to fulfill the distribution request are on disk, and ready to be copied. The correct archive object to request is determined from the information provided by the SDSRV in the distribution request. This returns references to the files in the read-only cache.
F.3.12	Link files to Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST links the files from the read-only cache into the staging disk.
F.3.13	Copy files to Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST copies the subsetted and metadata files from the SDSRV’s Staging Disk into the staging disk.
F.4.1	Allocate Media Resource	EcDsDdRequestMgrMain	EcDsSt8mmTapeServer	Distributed Object	DDIST now creates the Resource manager for 8mm via a Resource Manager Factory. The correct resource manager is determined from the MediaType handed to the resource factory (8mm, in this case). The correct 8mm resource is determined from configuration within the resource factory.

Table 3.4.9.3-1. Component Interaction Table: Landsat Scenario, Ordering WRS Scenes (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
F.4.2	Update Tracked Request Status	EcDsDdRequestMgrMain	EcAcOrderManager	Distributed Object	Update the status of the tracked request to "Transferring".
F.4.3	Update Tracked Status	EcAcOrderManager	Sybase	Ctlib	Update the Order Tracking Database with new status.
F.4.4	Write files to 8mm tape	EcDsDdRequestMgrMain	EcDsSt8mmTapeServer	Distributed Object	DDIST requests that the tapes held in staging disk be copied to the 8mm device. Upon completion of the files being copied, the state of the distribution request is marked as "Ready for Shipment", which is displayed on the operator GUI. A packing list is generated.
F.4.5	Update Tracked Request Status	EcDsDdRequestMgrMain	EcAcOrderManager	Distributed Object	Update the status of the tracked request to "Waiting for Shipment".
F.4.6	Update Tracked Status	EcAcOrderManager	Sybase	Ctlib	Update the Order Tracking Database with new status.
F.5.1	Determine request is ready for shipping	DAAC Ops - Distribution Technician	EcDsDdistGui	GUI	Operator notes that the request is ready for shipping and that it includes the 8mm tape just produced. The 8mm tape slot and stacker id are included on the request screen, so that the operator knows which tapes to ship.
F.5.2	Ship Tapes	DAAC Ops - Data Technician			Labels for all media, as well as a shipping label for the package are created manually. Using commercial shipping vendors (determined by DAAC policy), the DAAC Data Technician labels the tape, packages the tape(s) and packing list, labels the package and ships to address provided with the request.
F.6.1	Mark as Shipped	DAAC Ops - Data Technician	EcDsDdistGui	GUI	Using the DSS GUI, the Data Technician marks the request as "Shipped".
F.6.2	Update Distribution Request	EcDsDdistGui	EcDsDdRequestMgrMain	Distributed Object	DDIST updates the state of the request to "Shipped".
F.6.3	Update Tracked Request Status	EcDsDdRequestMgrMain	EcAcOrderManager	Distributed Object	Update the status of the tracked request to "Shipped".
F.6.4	Update Tracked Status	EcAcOrderManager	Sybase	Ctlib	Update the Order Tracking Database with new status.
F.7.1	Build Distribution Notice	EcDsDdRequestMgrMain			The DDIST builds an email notification that the user's order has been fulfilled. This notification will include the media id, type and format, as well as UR, type and file names and sizes for each granule.

Table 3.4.9.3-1. Component Interaction Table: Landsat Scenario, Ordering WRS Scenes (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
F.7.2	Send E-mail	EcDsDdRequestMgrMain	Science User	email	Message is email'd to the Science User's email address, determined from the UserProfile.
F.8.1	Select data to be ordered via ftpPull	Science User	xims	GUI	The Science User selects a specific WRS scenes to order for delivery via ftpPull while the tape order is being processed. When the order is complete it is submitted to the V0 Gateway.
F.9.1	V0 Gateway Order	xims	EcDmV0ToEcsGateway	ODL, over sockets	B0SOT submits an order to the V0 Gateway, by converting the order into an ODL structure and passing that structure to a socket provided by the Gateway. The correct socket is determined from configuration information in the Valids file.
F.9.2	Establish ECS User	EcDmV0ToEcsGateway	MsAcManager	Distributed Object	V0 Gateway retrieves the User Profile using ECS Authenticator from ODL message, which includes an encrypted UserID and Password. The User Registration Server is replicated across DAACs, so the connection is made to the local User Registration Server.
F.9.3	Translate Terms	EcDmV0ToEcsGateway	EcDmDictServer	Ctlib (RWDbTool)	Gateway translates the V0 terms from ODL into ECS names for request submittal. Interface currently is directly to Data Dictionary database. Database name is retrieved from configuration file.
F.9.4	Create Tracked Order	EcDmV0ToEcsGateway	MsAcOrderManager	Distributed Object	Create an order to be tracked within ECS. Initial status set to "Pending".
F.9.5	Persist Tracked Order	EcAcOrderManager	Sybase	Ctlib	Create a tracked order in the database.
F.9.6	Create Tracked Request	EcDmV0ToEcsGateway	MsAcOrderManager	Distributed Object	Create the Tracked Request within the Order. Initial status set to "Pending".
F.9.7	Persist Tracked Request	EcAcOrderManager	Sybase	Ctlib	Create a tracked request in the order database.
F.9.8	Connect to SDSRV	EcDmV0ToEcsGateway	EcDsScienceDataServer	Distributed Object	The Gateway first connects to the SDSRV. The correct SDSRV is determined by the UR of a granule that is being ordered.
F.9.9	Add granules to Session	EcDmV0ToEcsGateway	EcDsScienceDataServer	Distributed Object	The V0 Gateway establishes the data context of the session with the SDSRV by adding the WRS Scene granule to the session. The GranuleUR of the granule to be ordered is added to the ESDTReferenceCollection.

Table 3.4.9.3-1. Component Interaction Table: Landsat Scenario, Ordering WRS Scenes (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
F.9.10	Retrieve Granule Metadata from Inventory	EcDsScienceDataServer	Sybase/SQS	CtLib	SDSRV completes establishing the data context by retrieving the metadata for the requested granules from the Sybase/SQS database. The metadata for each granule is passed back to the reference objects for each granule.
F.10.1	Acquire Data	EcDmV0ToEcsGateway	EcDsScienceDataServer	Distributed Object	V0Gateway submits the order for WRS Scene granules by submitting an Acquire request for the granules. The Acquire request is for an ftpPull of a specified granule in the ESDTReferenceCollection. This request is asynchronous, meaning that the return of the submit call of the request will only contain the status of the request's submittal. The Acquire request structure was determined from the Action submitted with the standing order method. For the granule referenced in the Acquire request, the SDSRV creates a file containing the granules metadata before passing to Distribution.
F.10.2	Create Staging Disk	EcDsScienceDataServer	EcStStagingDiskServer	Distributed Object	SDSRV creates Staging Disk for working space, scene files and metadata files, which allocates space and passes back a reference to that disk space. The reference to the Staging Disk is determined from the ArchiveID metadata of the granule to be distributed The amount of staging disk to request is determined on a band-by-band basis. The SDSRV requests twice the size of the first band, from metadata in the database, and requests more as needed in order to subset the subsequent files. Each request is for twice the size of the file to be subsetted.
F.10.3	STMGT Retrieve	EcDsScienceDataServer	EcDsStArchiveServer	Distributed Object	SDSRV requests that STMGT retrieve the subinterval granule files that are to be subsetted. For a WRS Scene, these files include band files, calibration files, MTA, MSCD and PCD files. This will result in the files being staged to the working staging disk area. The correct archive object to request is determined from the information provided by the SDSRV in the distribution request.
F.10.4	L7 Scene creation	EcDsScienceDataServer			Since L7 WRS Scenes are virtual granules (i.e. the granules are represented in the inventory with their metadata, but the files which contain the data don't actually exist), the scene files must be extracted from the scene's parent subinterval. The scenes files are created using internal subinterval subsetting methods. The subsetted files include band files, calibration files, MTA, MSCD, PCD, and CPF files. A metadata file for the WRS Scene is created.

Table 3.4.9.3-1. Component Interaction Table: Landsat Scenario, Ordering WRS Scenes (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
F.10.5	Distribute Granules, Synchronous	EcDsScienceDataServer	EcDsDdRequestMgrMain	Distributed Object	SDSRV submits a request to Data Distribution. The request includes, for the granule, a reference to the metadata file as well as the data file. Other parameters from the Acquire request are passed to DDIST.
F.10.7	Update Tracked Request Status	EcDsDdRequestMgrMain	MsAcOrderManager	Distributed Object	Update the status of the tracked request to “Active”.
F.10.8	Update Tracked Status	EcAcOrderManager	Sybase	Ctlib	Update the Order Tracking Database with new status.
F.10.9	Create Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST creates Staging Disk for the granule files in the archive. This allocates space and passes back a reference to that disk space. The reference to the Staging Disk is determined from the information passed by the SDSRV in the distribution request ,which was the ArchiveID metadata parameter of the granule to be staged. The amount of staging disk to request is calculated from the file sizes in the information passed in the Distribution Request.
F.10.9	Update Tracked Request Status	EcDsDdRequestMgrMain	MsAcOrderManager	Distributed Object	Update the status of the tracked request to “Staging”.
F.10.10	Update Tracked Status	EcAcOrderManager	Sybase	Ctlib	Update the Order Tracking Database with new status.
F.10.11	STMGT Retrieve	EcDsDdRequestMgrMain	EcDsStArchiveServer	Distributed Object	DDIST requests that STMGT retrieve the granule files that are still archived. This will result in the file being staged to read-only cache disks. This means that all files needed to fulfill the distribution request are on disk, and ready to be copied. The correct archive object to request is determined from the information provided by the SDSRV in the distribution request. This returns references to the files in the read-only cache.
F.10.12	Link files to Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST links the files from the read-only cache into the staging disk.
F.10.13	Copy files to Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST copies the subsetted and metadata files from the SDSRV’s Staging Disk into the staging disk.
F.10.14	Update Tracked Request Status	EcDsDdRequestMgrMain	MsAcOrderManager	Distributed Object	Update the status of the tracked request to “Transferring”.
F.10.15	Update Tracked Status	EcAcOrderManager	Sybase	Ctlib	Update the Order Tracking Database with new status.

Table 3.4.9.3-1. Component Interaction Table: Landsat Scenario, Ordering WRS Scenes (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
F.10.16	DDIST Pull No Tar	EcDsDdRequestMgrMain	EcDsStFtpDisServer	Distributed Object	DDIST now creates the Resource manager for ftpPull via a Resource Manager Factory. The correct resource manager is determined from the MediaType handed to the resource factory (ftpPull, in this case). The correct FtpServer is determined from configuration within the resource factory.
F.10.17	Update Tracked Request Status	EcDsDdRequestMgrMain	MsAcOrderManager	Distributed Object	Update the status of the tracked request to “Shipped”
F.10.18	Update Tracked Status	EcAcOrderManager	Sybase	Ctlib	Update the Order Tracking Database with new status.
F.11.1	Build Distribution Notice	EcDsDdRequestMgrMain			DDIST builds a distribution notice indicating that the WRS Scene data is now available. The notice includes the UR of the Scene granule, name of the Scene’s files, the host and directory name where it is available and how long it will be available.
F.11.2	Send E-mail	EcDsDdRequestMgrMain	Science User		The distribution notice is email’d to the Science User.

3.5 ASTER Scenario

This scenario shows how the ECS supports the ASTER mission. ECS provides a mechanism for ECS Users to submit Data Acquisition Requests (DARs). ECS notifies the ECS User when that DAR has been fulfilled. ECS receives ASTER data via tape, from ASTER GDS. These tapes contain L1A and L1B data. This data is provided to ECS regardless of whether or not ECS Users had previously submitted DARs. ECS provides support for users to request processing of the L1A and L1B data to higher information levels, via requests for On-Demand Processing. A request for On-Demand Processing may required a sequence of algorithms be run on the specified data. Granules produced by On-Demand Processing are not permanently archived. ECS also supports the insertion of ASTER Expedited Data Set (EDS) from EDOS, and its immediate availability to selected ASTER Scientists.

For purposes of the demonstration, the ASTER DAR Comm Gateway did not actually interface to ASTER’s GDS. Instead, the RPC-level calls that the ASTER DAR Comm Gateway would normally make to ASTER GDS software, were implemented with internal ASTER DAR Comm Gateway logic. The behavior of these implementations was designed to simulate the actual ASTER GDS behavior. This scenario includes the workaround for ASTER backward chaining and On-Demand Processing. Also, this scenario provides a simplified version of ASTER Expedited Data support. In this simplified version, ECS simply notifies, via email, the Instrument Team as to the availability of new ASTER Expedited data.

The following system functionality is exercised in this scenario:

- DARTool usage for DAR submittal
- Data Tape Ingest
- Backward Chaining workaround
- SCF metadata update workaround
- Simplified ASTER Expedited Data Support

The following diagram illustrates the relationships between the data types and PGEs used in the ASTER scenario:

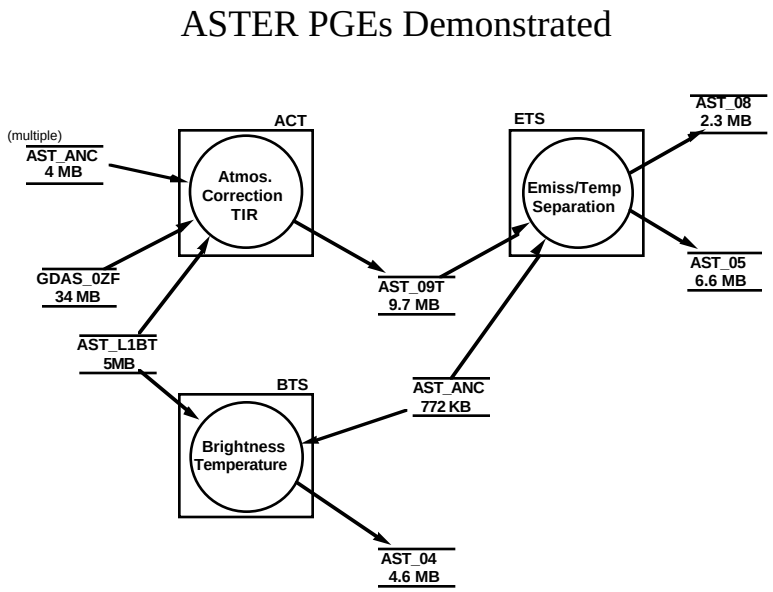

Figure 3.5-1. ASTER Scenario PGE/Data Relationships

3.5.1 Scenario Preconditions

The following ESDTs have been inserted into the ECS:

AST_ANC (ASTER Ancillary data)
AST_EXP (ASTER Expedited L0 data)
AST_L1BT (Test version of ASTER L1B TIR data)
AST_04 (L2 Brightness Temperature)
AST_05 (L2 Surface Emissivity)
AST_08 (L2 Surface Temperature)
AST_09T (L2 Surface Radiance TIR)
GDAS_0ZF (NCEP provided ancillary data)
PH (Product History)
PGEEXE (PGE Execution Granule)

The following ASTER PGEs have passed SSI&T and have been inserted into the ECS:

ACT
ETS
BTS

Ancillary granules (AST_ANC and GDAS_0ZF) have been inserted into the ECS.

3.5.2 Scenario Partitions

The ASTER scenario has been partitioned into the following threads:

DAR Submission (Thread A) - This thread will show the usage of the DARTool, and its interaction with GDS (which is simulated) and other ECS components.

GDS Tape Insertion (Thread B) - This thread will show how the ECS inserts data provided by GDS on D3 tape.

Backward Chaining (Thread C) - This thread will show how the system supports requests from ECS users to produce data requiring a sequence of algorithms to be run.

QA Metadata Update (Thread D) - This thread will show how the ECS supports updating the QA metadata of a specified granule.

On-Demand Production (Thread E) - This thread will show how the ECS supports users request for On-Demand production.

Simplified Expedited Data Support (Thread F) - This thread will show how the ECS supports a simplified version of Expedited data support.

3.5.3 DAR Submission Thread

3.5.3.1 Interaction Diagram - Domain View

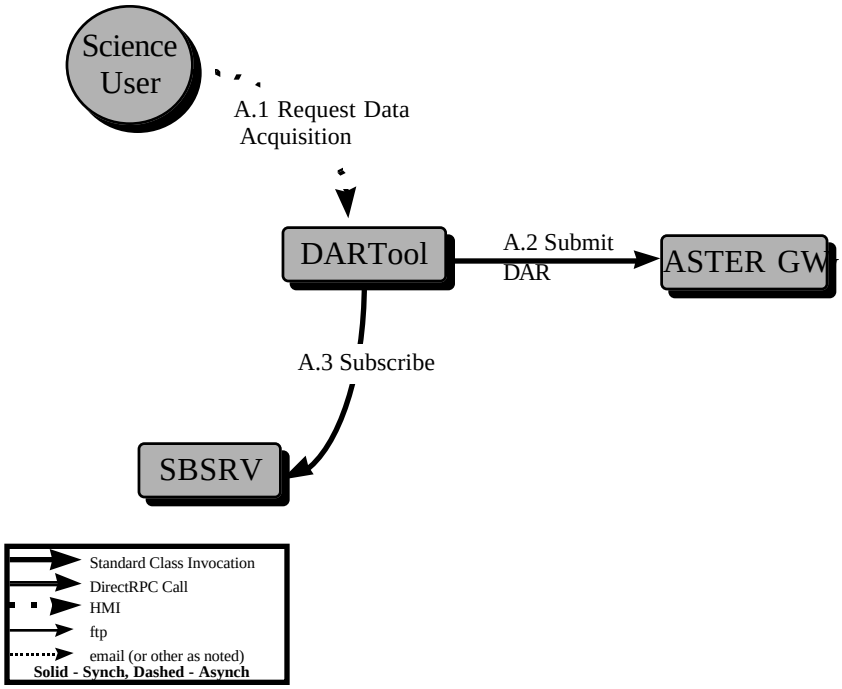

Figure 3.5.3.1-1. DAR Submission Interaction Diagram

3.5.3.2 Interaction Table - Domain View

Table 3.5.3.2-1. Interaction Table - Domain View: ASTER Scenario, DAR Submission

Step	Interaction	Interface Client	Interface Provider	Data Issues	Preconditions	Description
A.1	Request Data Acquisition	Science User	DARTool		Currently DARTool only available from DAAC.	Scientist uses DARTool to request ASTER data take(s). Scientist inputs, at a minimum, the location of the desired data take. Please note that there is no guarantee that an accepted Data Acquisition Request will actually result in an ASTER data take, nor is there any guarantee of when the data take may occur.
A.2	Submit DAR	DARTool	ASTER GW			DARTool submits the request for the ASTER data take(s) to the ASTER DAR Comm Gateway. Normally, the ASTER DAR Comm Gateway passes the request to ASTER's GDS, which responds with a XARid. Currently, the interface between the ASTER DAR Comm Gateway and GDS is implemented completely within the ASTER DAR Comm Gateway, by stubbed versions of the api's provided by GDS. The ASTER DAR Comm Gateway still responds with a XARid.
A.3	Subscribe	DARTool	SBSRV			The DARTool places a qualified subscription on behalf of the Science User to be notified when AST_L1B granules, with a XARid matching the desired XARid, are inserted.

3.5.3.3 Component Interaction Table

Table 3.5.3.3-1. Component Interaction Table: ASTER Scenario - DAR Submission

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
A.1.1	Startup DARTool	Science User	DART	Xevent	Science User invokes the DARTool GUI Application. The Science User has already been logged on the Science Desktop and begins DARTool by double-clicking on an icon. Upon startup the Desktop sends the User ID to the DART.
A.1.2	Specify Acquisition characteristics	Science User	DART	GUI	The Science User provides appropriate input to specify the desired data take and any sensor attributes.
A.2.1	Submit DAR	DART	EcGwDAR Server	rpc	DARTool forwards request information to the ASTER DAR Comm Gateway, for submittal to ASTER GDS. The correct DAR Gateway is determined by configuration, as there is only one DAR Gateway in ECS, which resides at EDC.
A.2.2	Submit DAR to GDS	EcGwDAR Server	ASTER GDS	sockets	ASTER DAR Comm Gateway establishes communication with GDS, and submits the DAR. GDS responds with the XARid, which is returned to the DARTool. Correct sockets are specified in the ECS-ASTER ICD.
A.3.1	Search for Service Advertisements	EcGwDAR Server	EcIoAdServer	Distributed Object	ASTER DAR Comm Gateway searches Advertiser for the service to use for Subscribe to AST_L1BT:Insert event. This is accomplished via the IoAdApprovedSearchCommand class. Since the Advertiser is based on a replicated database, no specification is required to select the proper Advertiser. The local one is used.
A.3.1	Connect to SBSRV	DART	EcSbSubServer	Distributed Object	The DARTool connects to the subscription server in order to subscribe to AST_L1BT granules that match the given XARid. This is a qualified subscription. The correct Subscription server is determined from the Subscribe Advertisement.
A.3.2	Submit Subscription	DART	EcSbSubServer	Distributed Object	Submit the subscription to the Subscription Server. This is accomplished with the EcClSubscription interface class.
A.3.3	Persist a Subscription	EcSbSubServer	Sybase	CtLib	Subscription is stored in the Sybase Database.

3.5.4 GDS Tape Insertion Thread

3.5.4.1 Interaction Diagram - Domain View

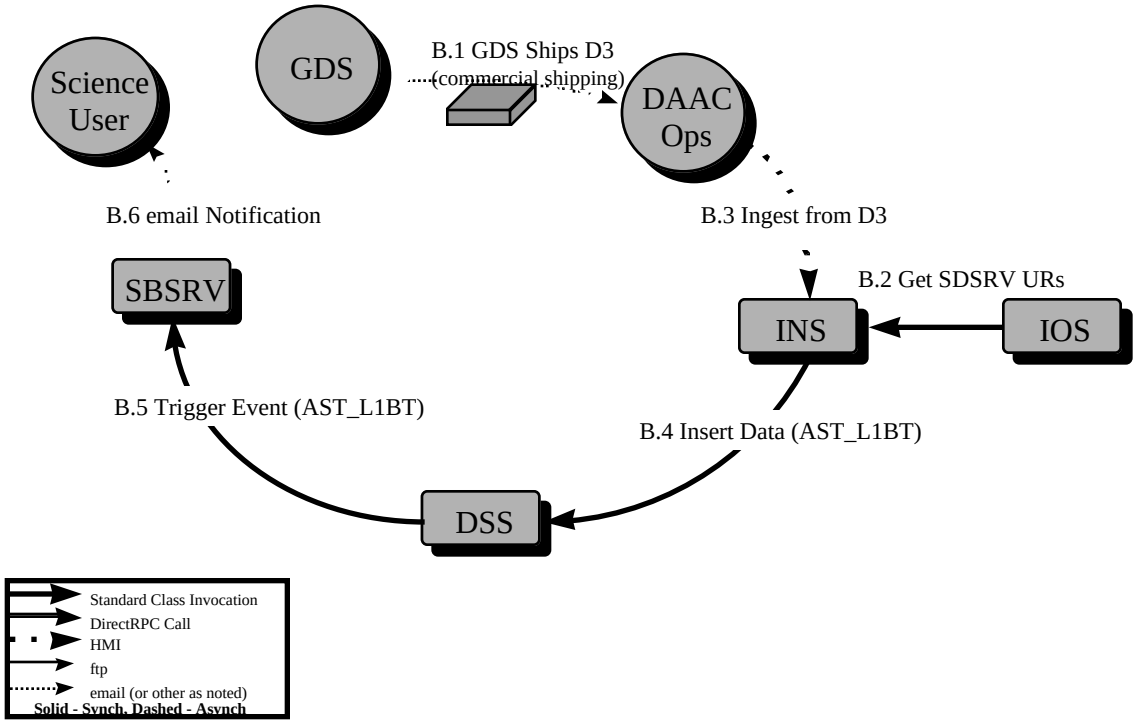

Figure 3.5.4.1-1. GDS Tape Insertion Interaction Diagram

3.5.4.2 Interaction Table - Domain View

Table 3.5.4.2-1. Interaction Table - Domain View: ASTER Scenario, GDS Tape Insertion

Step	Interaction	Interface Client	Interface Provider	Data Issues	Preconditions	Description
B.1	GDS Ships D3 Tape	GDS	DAAC Ops			GDS uses commercial shipping vendor, sends a D3 tapes containing AST_L1A or AST_L1B data. Tape contains data takes that are both due to ECS DARs as well as data that wasn't requested via ECS.
	Get SDSRV URs	INS	IOS			Upon startup, Ingest gets the SDSRV URs for each data type in its database.
B.2	Ingest from D3	DAAC Ingest/Distribution Technician	INS			DAAC Ingest/Distribution Technician loads the D3 tape and, using an Ingest GUI, begins the reading of data granules from the tape.
B.3	Insert Data	INS	DSS	2 granules @ 5 MB	AST_L1BT ESDT	Ingest inserts the new ASTER granules into the Data Server.
B.4	Trigger Event	DSS	SBSRV			Upon successful completion of insertion of each AST_L1BT granule, the AST_L1BT:Insert event is triggered, with the qualifiers including all the XARids attached to that data.
B.5	Notification	SBSRV	Science User			Send notification to Science User that AST_L1BT granules for their DAR have been inserted. Notification message includes the UR of the granule, as well as the DAR ids that have been matched.

3.5.4.3 Component Interaction Table

Table 3.5.4.3-1. Component Interaction Table: ASTER Scenario - GDS Tape Insertion (1 of 2)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.2.1	Startup Media Ingest GUI	DAAC Ingest Technician	EcInGUI	Xevent	DAAC Ingest Technician invokes the Ingest Media GUI. While already running within the DAAC Desktop, the DAAC Ingest Technician double clicks on the Ingest GUI icon.
B.2.2	Select Ingest Device	DAAC Ingest Technician	EcInGUI	GUI	DAAC Ingest Technician selects the media device (D3) to read data from.
	Get SDSRV URs from IOS	EcInReqMgr	EcIoAdServer	Distributed Object	Upon startup, Ingest Request Manager requests the SDSRV URs for each data type in its database.
B.2.3	Allocate Media Resource	EcInGUI	EcDsStD3TapeServer	Distributed Object	Ingest now creates the Resource manager for D3 via a Resource Manager Factory. The correct D3 resource is determined from configuration within the resource factory.
	Create Staging Disk	EcInGUI	EcStStagingDiskServer	Distributed Object	Ingest creates a Staging Disk for the delivery record file.
B.2.4	Read D3 Tape	EcInGUI	EcDsStD3TapeServer	Distributed Object	Ingest reads the delivery record file. From this file the type and amount of data to be read is determined. The delivery record file is in the first tar set on the tape.
B.2.5	Create Staging Disk	EcInGUI	EcStStagingDiskServer	Distributed Object	Ingest creates Staging Disk. The reference to the Staging Disk server is determined from the Ingest Database. The amount of staging disk to request is determined from the delivery record file.
B.2.4	Read D3 Tape	EcInGUI	EcDsSt8mmTapeServer	Distributed Object	Ingest reads data files from the D3 tape.
B.2.5	Ingest Request	EcInGUI	EcInReqMgr	Distributed Object	Ingest GUI process packages the files read into the appropriate Ingest Requests. The data source (D3 tape), defined on startup, is passed to the Ingest Request Manager.
	Ingest Granule	EcInReqMgr	EcInGran	Distributed Object	Request Manager packages the request into granules and sends them to the Ingest Granule Server.
B.3.1	Connect to SDSRV	EcInGran	EcDsScienceDataServer	Distributed Object	Ingest begins a session with the SDSRV by connecting. The correct SDSRV is determined during EcInReqMgr startup, from Advertising, based on the data type. The data type is determined from the delivery record file.
B.3.2	Get Metadata Configuration File	EcInGran	EcDsScienceDataServer	Distributed Object	Ingest requests the metadata configuration file (mcf) for the data being inserted.

Table 3.5.4.3-1. Component Interaction Table: ASTER Scenario - GDS Tape Insertion (2 of 2)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
B.3.3	Validate Metadata	EcInGran	EcDsScienceDataServer	Distributed Object	After building a metadata file for the input data granule, Ingest asks SDSRV to validate the metadata, based on the granule's data type.
B.3.4	Insert Data	EcInGran	EcDsScienceDataServer	Distributed Object	Ingest requests that the received files for the data granule are inserted into the Data Server. An Insert request, containing the names of the files comprising the granule, is created. The structure of the Insert Request is hardcoded in the granule server. SDSRV re-validates metadata and determines the archived names of the files. Upon completion of the insert, the status is asynchronously reflected on the GUI screen.
B.3.5	STMGT Store	EcDsScienceDataServer	EcDsStArchiveServer	Distributed Object	SDSRV requests that the granule's files are archived. The archive server reads the inserted files directly from the Ingest staging disks that they are residing on. The correct archive object to request is determined from collection level metadata for the ESDT, defined in the ESDT's descriptor.
B.3.6	Adding a Granule to Inventory	EcDsScienceDataServer	Sybase/SQS	CtLib	The validated metadata is parsed and added to the inventory of the SDSRV.
B.4.1	Trigger Event	EcDsScienceDataServer	EcSbSubServer	Distributed Object	Upon successful insertion of data granule, the AST_L1BT:Insert event is triggered. The correct subscription server is determined from SDSRV configuration. The correct events to trigger are determined from the events file, which was populated during ESDT installation. Provided with the event triggering is the UR of the inserted granule.
B.4.2	Fire Subscriptions	EcSbSubServer	Sybase	CtLib	SBSRV queries the Sybase database determining which subscriptions need to be activated, or fired. Each query "hit" is an activated subscription and will execute independently.
B.5.1	Send E-mail	EcSbSubServer	Science User		The SBSRV builds an email notification that the user's subscription on the AST_L1BT:Insert event has been fired. This notification will identify the event, the subscription ID, the GranuleUR that was inserted and the previously supplied UserString.

3.5.5 Backward Chaining Thread

3.5.5.1 Interaction Diagram - Domain View

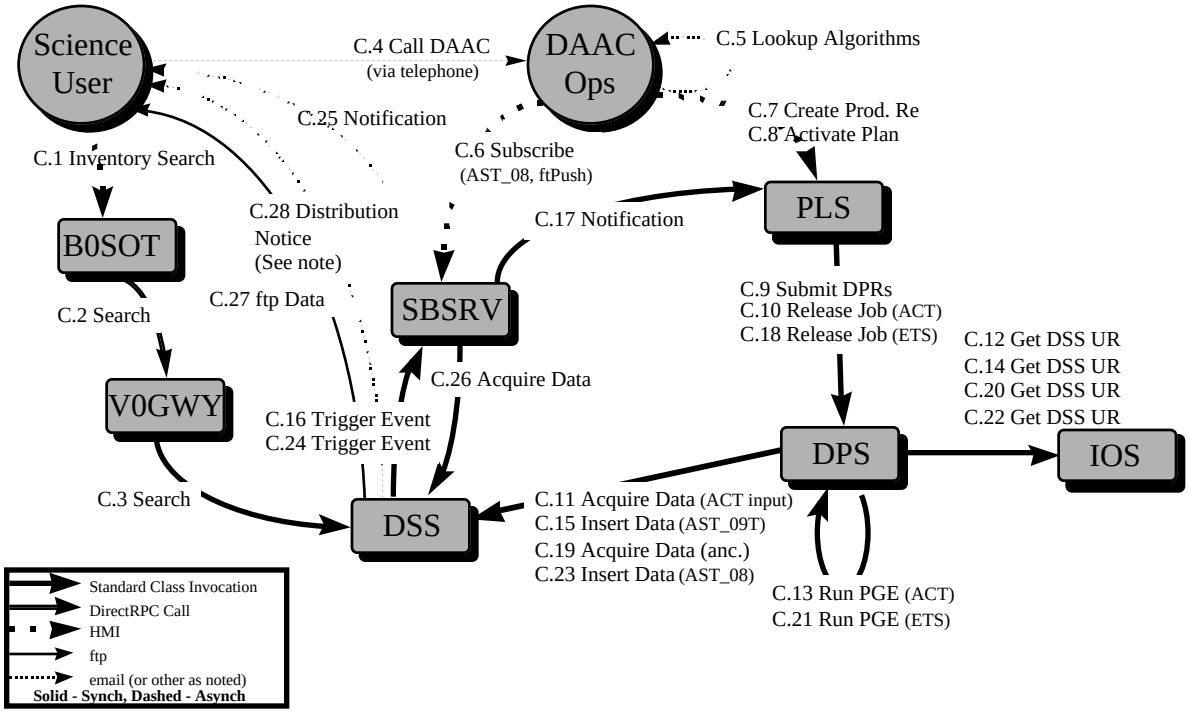

Figure 3.5.5.1-1. Backward Chaining Interaction Diagram

3.5.5.2 Interaction Table - Domain View

Table 3.5.5.2-1. Interaction Table - Domain View: ASTER Scenario, Backward Chaining

Step	Interaction	Interface Client	Interface Provider	Data Issues	Preconditions	Description
C.1	Inventory Search	Science User	B0SOT			Upon notification that the data resulting from the DAR, the Science User looks for up the data granule, in order to determine it's metadata characteristics.
C.2	Search	B0SOT	GTWAY			B0SOT submits the Science User's search criteria to the V0 Gateway in ODL format, via a well defined socket.
C.3	Search	GTWAY	SDSRV			The V0 gateway translates the Search criteria from ODL to a query object (using GIParameters), and submits that query to the Search service. The results of this Search are returned synchronously, and are passed back to B0SOT, which displays them to the Science User.
C.4	Call DAAC	Science User	DAAC User Services Representative			Upon determining that the data take resulted in useful data, the Scientist decides to call the DAAC, requesting that a L2 Surface Temperature (AST_08) granule be produced from the AST_L1BT data. The Scientist request that the AST_08 data be shipped electronically to his/her workstation.
C.5	Lookup Algorithms	DAAC Production Planner	Technical Baseline			The DAAC Production Planner determines the process to take the AST_L1BT data into AST_08 data. The process is a two-stage algorithm sequence: chaining the ACT and ETS algorithms.
C.6	Subscribe	DAAC Production Planner	SBSRV			The DAAC Production Planner places a subscription for the Science User to receive the resultant AST_08 granule, via an ftpPush.
C.7	Create Production Request	DAAC Production Planner	PLS			DAAC Production Planner creates DPRs for ACT and ETS PGEs.
C.8	Activate Production Plan	DAAC Production Planner	PLS		PGEs passed SSI&T. Plan already created.	DAAC Production Planner activates a plan, which includes DPRs for ACT and ETS PGEs.
C.9	Submit DPRs	PLS	DPS			DPRs for ACT and ETS are submitted to DPS.

Table 3.5.5.2-1. Interaction Table - Domain View: ASTER Scenario, Backward Chaining (Continued)

Step	Interaction	Interface Client	Interface Provider	Data Issues	Preconditions	Description
C.10	Release Job	PLS	DPS			Since all inputs are available to run the ACT PGE, references to those input granules are passed to DPS, and the ACT job is released.
C.11	Acquire Data	DPS	DSS	24 AST_ANC @4MB, 1 GDAS_0ZF @4MB, 1 AST_L1BT @5 MB	AST_ANC & GDAS_0ZF data already inserted	DPS submits Acquire Request for input granules, via ftpPush, for input to ACT.
C.12	Get DSS UR	DPS	IOS			DPS gets the DSS UR from Advertiser.
C.13	Run PGE			1 AST_09T @9.7 MB produced		ACT runs, creating AST_09T granules.
C.14	Get DSS UR	DPS	IOS			DPS gets the DSS UR from Advertiser.
C.15	Insert Data	DPS	DSS		AST_09T ESDT	Archive newly created AST_09T granule.
C.16	Trigger Event	SDSRV	SBSRV			Trigger AST_09T:Insert event.
C.17	Notification	SBSRV	PLS		PLS Subscriptions for AST_09T:Insert event	Send direct notification to PLS, notifying that there is a newly inserted AST_09T granule. Notification message includes the UR of the AST_09T granule.
C.18	Release Job	PLS	DPS			PLS releases job containing ETS.
C.19	Acquire Data	DPS	DSS	1 AST_ANC @722MB	AST_ANC data already inserted	DPS submits Acquire Request for the ancillary product, AST_ANC, via ftpPush, for input to ETS. Note that other input to ETS, AST_09T. is already available on DPS resources.
C.20	Get DSS UR	DPS	IOS			DPS gets the DSS UR from Advertiser.
C.21	Run PGE			1 AST_08 @2.3 MB, 1 AST_09T @6.6MB		ETS runs, creating both AST_08 and AST_05 data granules.
C.22	Get DSS UR	DPS	IOS			DPS gets the DSS UR from Advertiser.

Table 3.5.5.2-1. Interaction Table - Domain View: ASTER Scenario, Backward Chaining (Continued)

Step	Interaction	Interface Client	Interface Provider	Data Issues	Preconditions	Description
C.23	Insert Data	DPS	SDSRV		AST_08 and AST_09T ESDTs	Archive newly created AST_08 and AST_05 granules.
C.24	Trigger Event	SDSRV	SBSRV			Trigger AST_08:Insert and AST_05:Insert events.
C.25	Notification	SBSRV	Science User			Send email notification to Science User, notifying that the AST_08 granule has been inserted. Notification message includes the UR of the AST_08 granule.
C.26	Acquire Data	SBSRV	SDSRV			SDSRV submits an Acquire Request, on behalf of the Science User, to have the AST_08 granule shipped, via ftpPush, to the Scientists workstation.
C.27	ftp Data	DSS	Science User	1 AST_08 @ 2.3 MB		DSS ftp's the AST_08 data to the Scientist's workstation.
C.28	Distribution Notice	DSS	Science User			Please note that this step does not actually occur, due to a system flaw. DSS emails notification to the Science User, notifying the presence of the AST_08 data on their workstation.

3.5.5.3 Component Interaction Table

Table 3.5.5.3-1. Component Interaction Table: ASTER Scenario - Backward Chaining

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
C.1.1	Startup B0SOT	Science User	xims	XEvent	Science User invokes the B0SOT GUI Application. The user has already been logged on the DAAC Desktop and begins B0SOT by double-clicking on an icon.
C.1.2	Select Inventory Search, Provide Query constraints, Submit Query	Science User	xims	GUI	The Science User provides search constraints for the AST_L1BT granules desired. When query constraints are completed, the query is submitted.
C.2.1	V0 Gateway Inventory	xims	EcDmV0ToEcsGateway	ODL, over sockets	B0SOT submits a search to the V0 Gateway, by converting the search criteria into an ODL structure and passing that structure to a socket provided by the Gateway. The correct socket is determined from configuration information contained in the Valids file.

Table 3.5.5.3-1. Component Interaction Table: ASTER Scenario - Backward Chaining (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
C.3.1	Establish ECS User	EcDmV0ToEcsGateway	MsAcManager	Distributed Object	V0 Gateway retrieves the User Profile using ECS Authenticator from ODL message, which includes an encrypted UserID and Password. The User Registration Server is replicated across DAACs, so the connection is made to the local User Registration Server.
C.3.2	Translate Terms	EcDmV0ToEcsGateway	EcDmDictServer	Ctlib (RWDbTool)	Gateway translates the V0 terms from ODL into ECS names for query submittal. Interface currently is directly to Data Dictionary database. Database name is retrieved from configuration file.
C.3.3	Connect to SDSRV	EcDmV0ToEcsGateway	EcDsScienceDataServer	Distributed Object	The Gateway first connects to the SDSRV. The correct SDSRV is determined from configuration information.
C.3.4	SDSRV Query	EcDmV0ToEcsGateway	EcDsScienceDataServer	Distributed Object	The Gateway translates the query into a DsClQuery object. This object is handed to the Search interface of the DsClESDTReferenceCollector. This Search method is synchronous, so the results of the search will be returned to the calling function.
C.3.5	Querying the Inventory	EcDsScienceDataServer	Sybase/SQS	CtLib	The SDSRV breaks down the Query object and translates it into a sequence of calls to the inventory database. Resultant rows are converted into data granules, each with their metadata extracted from the database. These results are packaged and returned to the Query client.
C.3.6	Result Retrieval	xims	EcDmV0ToEcsGateway	ODL, over Sockets	When the Gateway gets the results, they are translated into ODL, and passed back to the B0SOT tool. The correct socket for sending results to B0SOT is the one used to submit the query. B0SOT then displays the results of the query to the user.
C.6.1	Startup SBSRV GUI	DAAC User Services Representative	EcSbGui	Xterm	After receiving a call from the user for AST_08 data made from the AST_L1BT granules, User Services Representative then calls DAAC Production Planner, communicating the need for the AST_08 product. DAAC Production Planner determines the sequence of algorithms required. The algorithms needed are determined from the Technical Baseline and a series of queries on the PDPS database?. DAAC User Services Representative invokes SBSRV GUI application.

Table 3.5.5.3-1. Component Interaction Table: ASTER Scenario - Backward Chaining (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
C.6.2	Create & Submit Subscription from GUI	DAAC User Services Representative	EcSbGui	Xterm	DAAC User Services Representative represents him/herself as the Science User. The DAAC Operator brings up the GUI and clicks button to create new subscription. A list of events is then displayed from which the op can choose to subscribe. DAAC Operator selects the AST_08:Insert Event for subscription. Only one action (besides notification), is available from the SBSRV at this time. FtpPush as a distribution mechanism is input via a GUI button. Other parameters required for FtpPush, including the Science User's host name, target directory, ftp user name, and ftp password, are input via the GUI.
C.6.3	Submit Subscription	EcSbGui	EcSbSubServer	Distributed Object	Submit the subscription to the Subscription Server. This is accomplished with the EcClSubscription interface class. The correct SBSRV is determined via a ServerUR, declared in configuration.
C.6.4	Persist a Subscription	EcSbSubServer	Sybase	CtLib	Subscription is stored in the Sybase Database.
C.7.1	Startup Production Request Editor	DAAC Productrion Planner	EcPlPREditor_IF	GUI	DAAC Planner invokes the Production Request Editor. While already running within the DAAC Desktop, the planner double clicks on the Planning Workbench icon.
C.7.2	Build Production Requests	DAAC Production Planner	EcPlPREditor_IF	GUI	DAAC Planner creates Production Requests for the ACT and ETS algorithms. Algorithm (ACT and ETS) is selected, along with the time domain of the output (and input) data. Dependency of ETS on ACT, based on ACT output, is established.
C.7.3	Search for Service Advertisements	EcPlPREditor_IF	EcIoAdServer	Distributed Object	In order to ensure that the correct input data is used for the ACT algorithm, the Editor searches Advertiser for the service the Subscribe to AST_L1BT:Insert event. This is accomplished via the IoAdApprovedSearchCommand class. Since the Advertiser is based on a replicated database, no specification is required to select the proper Advertiser. The local one is used. Note: This (and the following SBSRV steps) only occurs if Planning does not already have a subscription for the AST_L1BT:Insert event.
C.7.4	Connect to SBSRV	EcPlPREditor_IF	EcSbSubServer	Distributed Object	The Editor connects to the subscription server in order to subscribe for notification of new AST_L1BT granules. The correct Subscription server is determined from the Subscribe Advertisement.
C.7.5	Submit Subscription	EcPlPREditor_IF	EcSbSubServer	Distributed Object	Submit the subscription to the Subscription Server. This is accomplished with the EcClSubscription interface class.

Table 3.5.5.3-1. Component Interaction Table: ASTER Scenario - Backward Chaining (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
C.7.6	Persist a Subscription	EcSbSubServer	Sybase	CtLib	Subscription is stored in the Sybase Database.
C.7.7	Search for Service Advertisements	EcPlPREditor_IF	EcIoAdServer	Distributed Object	In order to determine where (which SDSRV) the input data (AST_L1BT) is located the Editor searches the Advertiser for a "GetQueriableParameters" service for the desired input data type. This is in lieu of searching for Product Advertisements. This is accomplished via the IoAdApprovedSearchCommand class. Since the Advertiser is based on a replicated database, no specification is required to select the proper Advertiser. The local one is used. Note: AST_ANC and GDAS_0ZF are considered "static" data, and have references stored after SSI&T.
C.7.8	Connect to SDSRV	EcPlPREditor_IF	EcDsScienceDataServer	Distributed Object	Looking for input granules for the ACT PGE, the Planning Workbench first connects to the SDSRV. The correct SDSRV is determined from the service provider on the GetQueriableParameters Advertisement.
C.7.9	SDSRV Query	EcPlWb	EcDsScienceDataServer	Distributed Object	The Workbench builds a DsClQuery object, looking for AST_L1BT granules that match space and time parameters. This object is handed to the Search interface of the DsClESDTReferenceCollector. This Search method is synchronous, so the results of the search will be returned to the calling function.
C.7.10	Querying the Inventory	EcDsScienceDataServer	Sybase/SQS	CtLib	The SDSRV breaks down the Query object and translates it into a sequence of calls to the inventory database. Resultant rows are converted into data granules, each with their metadata extracted from the database. These results are packaged and returned to the Query client. Results are packaged in the ACT DPR.
C.7.11	Inspect Granule Value Parameters	EcPlPREditor_IF	EcDsScienceDataServer	Distributed Object	Editor checks the granule's metdata attributes (type, version, filesize and temporal range), to establish job dependencies.
C.7.12	Search for Service Advertisements	EcPlPREditor_IF	EcIoAdServer	Distributed Object	In order to ensure that the correct input data is used for the ETS algorithm, the Editor searches Advertiser for the service the Subscribe to AST_09:Insert event. This is accomplished via the IoAdApprovedSearchCommand class. Since the Advertiser is based on a replicated database, no specification is required to select the proper Advertiser. The local one is used. Note: This (and the following SBSRV steps) only occurs if Planning does not already have a subscription for the AST_09:Insert event.

Table 3.5.5.3-1. Component Interaction Table: ASTER Scenario - Backward Chaining (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
C.7.13	Connect to SBSRV	EcIoAdServer	EcSbSubServer	Distributed Object	The Editor connects to the subscription server in order to subscribe for notification of new AST_09 granules. The correct Subscription server is determined from the Subscribe Advertisement.
C.7.14	Submit Subscription	EcIoAdServer	EcSbSubServer	Distributed Object	Submit the subscription to the Subscription Server. This is accomplished with the EcClSubscription interface class.
C.7.15	Persist a Subscription	EcSbSubServer	Sybase	CtLib	Subscription is stored in the Sybase Database.
C.8.1	Startup Planning Workbench	DAAC Operator - Planner	EcPlWb	GUI	DAAC Planner invokes the Planning workbench. While already running within the DAAC Desktop, the planner double clicks on the Planning Workbench icon.
C.8.2	Create a Plan	DAAC Operator - Planner	EcPlWb	GUI	Planner interacts with Planning Workbench GUI to create a plan with DPRs for the ACT and ETS PGEs.
C.8.3	Activate a Plan	EcPlWb	EcDpPrJobMgmt	rpc	Updated plan is activated, making it the current processing plan.
C.9.1	Submit DPRs	EcDpPrJobMgmt	AutoSys	rpc (COTS)	The DPRs (one for ACT and a dependent one for ETS PGE) in plan are submitted, to AutoSys, by DPS for dependent execution. These jobs are dependent on input data.
C.10.1	Force Start Job	EcDpPrJobMgmt	event_demon	rpc	Job containing ACT is released.
C.10.2	Job Processing	event_demon	EcDpPrEM	command line	The job containing the ACT begins processing.
C.10.3	Connect to SDSRV	EcDpPrEM	EcDsScienceDataServer	Distributed Object	Processing begins a session with the SDSRV by connecting, in order to acquire the ACT PGE. The correct SDSRV is determined by using the GranuleUR of the PGE granule, which is defined in the Production plan and is part of the DPR.
C.10.4	Add granules to Session	EcDpPrEM	EcDsScienceDataServer	Distributed Object	PRONG establishes the data context of the session with the SDSRV by adding PGE granule's UR. The GranuleUR of the PGE granule is added to the ESDTReferenceCollection.
C.10.5	Retrieve Granule Metadata from Inventory	EcDsScienceDataServer	Sybase/SQS	CtLib	SDSRV completes establishing the data context by retrieving the metadata for the requested PGE granule from the Sybase/SQS database. The metadata for the PGE granule is passed back to the reference objects for each granule.

Table 3.5.5.3-1. Component Interaction Table: ASTER Scenario - Backward Chaining (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
C.10.6	Acquire Data	EcDpPrEM	EcDsScienceDataServer	Distributed Object	DPS requests granules by submitting an Acquire request for the PGE granule. The Acquire request is for an ftpPush of all granules in the ESDTReferenceCollection. This request is synchronous, meaning that the return of the submit call of the request will contain the results of the request. This means that the response will not be sent until the PGE granule files have been ftp'd to the DPS disks. This request asks for no distribution notice to be email'd. The Acquire request structure is hardcoded.
C.10.7	Create Staging Disk	EcDsScienceDataServer	EcStStagingDiskServer	Distributed Object	SDSRV creates Staging Disk for the metadata file, which allocates space and passes back a reference to that disk space. The reference to the Staging Disk is determined from the ArchiveID metadata of the granule to be distributed. The amount of staging disk to request is a default, hardcoded as a SDSRV constant.
C.10.8	Create Metadata file	EcDsScienceDataServer			The SDSRV creates a file containing the PGE granule's metadata before passing to Distribution.
C.10.9	Distribute Granules, Synchronous	EcDsScienceDataServer	EcDsDdRequestMgrMain	Distributed Object	SDSRV submits a request to DataDistribution. The request includes, for each granule, a reference to the metadata file as well as all data files. Other parameters from the Acquire request are passed to DDIST.
C.10.10	Create Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST creates Staging Disk for the granule files in the archive. This allocates space and passes back a reference to that disk space. The reference to the Staging Disk is determined from the information passed by the SDSRV in the distribution request ,which was the ArchiveID metadata parameter of the granule to be staged. The amount of staging disk to request is calculated from the file sizes of archived files, in the information passed in the Distribution Request.
C.10.11	STMGT Retrieve	EcDsDdRequestMgrMain	EcDsStArchiveServer	Distributed Object	DDIST requests that STMGT retrieve the PGE granule file that is archived. This will result in the file being staged to read-only cache disks. This means that all files needed to fulfill the distribution request are on disk, and ready to be copied. The correct archive object to request is determined from the information provided by the SDSRV in the distribution request. This returns references to the files in the read-only cache.
C.10.12	Link files to Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST links the files from the read-only cache into the staging disk.

Table 3.5.5.3-1. Component Interaction Table: ASTER Scenario - Backward Chaining (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
C.10.13	Copy files to Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST copies the metadata files from the SDSRV's Staging Disk into the staging disk.
C.10.14	ftpPush Files	EcDsDdRequestMgrMain	EcDsStFtpDisServer	Distributed Object	DDIST now creates the Resource manager for ftp Pushes via a Resource Manager Factory. The correct resource manager is determined from the MediaType handed to the resource factory (ftpPush, in this case). The correct FtpServer is determined from configuration within the resource factory. The files, host, location, username and password are all determined from the information provided in the original Acquire request.
C.10.15	ftp Files	EcDsStFtpDisServer	EcDpPrEM	rpc	CSS performs the actual low level ftp of the PGE files.
C.11.1	Connect to SDSRV	EcDpPrDM	EcDsScienceDataServer	Distributed Object	PRONG begins a session with the SDSRV by connecting. The correct SDSRV is determined by using the GranuleUR of the input granule.
C.11.2	Add granules to Session	EcDpPrDM	EcDsScienceDataServer	Distributed Object	PRONG establishes the data context of the session with the SDSRV by adding the input granules (1 AST_L1BT, 1 GDAS_0ZF and 24 AST_ANC) to the session. The GranuleUR of the input granule is added to the ESDTReferenceCollection. Note that this sequence is performed for each input granule, one at a time.
C.11.3	Retrieve Granule Metadata from Inventory	EcDsScienceDataServer	Sybase/SQS	CtLib	SDSRV completes establishing the data context by retrieving the metadata for the requested granule from the Sybase/SQS database. The metadata for each granule is passed back to the reference objects for each granule.
C.11.4	Acquire Data	EcDpPrDM	EcDsScienceDataServer	Distributed Object	DPS requests granules by submitting an Acquire request for those granules. The Acquire request is for an ftpPush of all granules in the ESDTReferenceCollection. This request is synchronous, meaning that the return of the submit call of the request will contain the results of the request. This means that the response will not be sent until the granule files have been ftp'd to the DPS disks. This request asks for no distribution notice to be email'd. The Acquire request structure is hardcoded.

Table 3.5.5.3-1. Component Interaction Table: ASTER Scenario - Backward Chaining (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
C.11.5	Create Staging Disk	EcDsScienceDataServer	EcStStagingDiskServer	Distributed Object	SDSRV creates Staging Disk for metadata files, which allocates space and passes back a reference to that disk space. The reference to the Staging Disk is determined from the ArchiveID metadata of the granule to be distributed. The amount of staging disk to request is determined from collection level metadata from the ESDT's Descriptor file.
C.11.6	Create Metadata file	EcDsScienceDataServer			For each granule referenced in the Acquire request, the SDSRV creates a file containing the granule's metadata before passing to Distribution.
C.11.7	Distribute Granules, Synchronous	EcDsScienceDataServer	EcDsDdRequestMgrMain	Distributed Object	SDSRV submits a request to DataDistribution. The request includes, for each granule, a reference to the metadata file as well as all data files. Other parameters from the Acquire request are passed to DDIST.
C.11.8	Create Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST creates Staging Disk for the granule files in the archive. This allocates space and passes back a reference to that disk space. The reference to the Staging Disk is determined from the information passed by the SDSRV in the distribution request ,which was the ArchiveID metadata parameter of the granule to be staged. The amount of staging disk to request is calculated from the file sizes in the information passed in the Distribution Request.
C.11.9	STMGT Retrieve	EcDsDdRequestMgrMain	EcDsStArchiveServer	Distributed Object	DDIST requests that STMGT retrieve the granule file that is archived. This will result in the file being staged to read-only cache disks. This means that all files needed to fulfill the distribution request are on disk, and ready to be copied. The correct archive object to request is determined from the information provided by the SDSRV in the distribution request. This returns references to the files in the read-only cache.
C.11.10	Link files to Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST links the files from the read-only cache into the staging disk.
C.11.11	Copy files to Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST copies the metadata files from the SDSRV's Staging Disk into the staging disk.

Table 3.5.5.3-1. Component Interaction Table: ASTER Scenario - Backward Chaining (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
C.11.12	ftpPush Files	EcDsDdRequestMgrMain	EcDsStFtpDisServer	Distributed Object	DDIST now creates the Resource manager for ftp Pushes via a Resource Manager Factory. The correct resource manager is determined from the MediaType handed to the resource factory (ftpPush, in this case). The correct FtpServer is determined from configuration within the resource factory. The files, host, location, username and password are all determined from the information provided in the original Acquire request.
C.11.13	ftp Files	EcDsStFtpDisServer	EcDpPrDM	rpc	CSS performs the actual low level ftp of the files.
C.12.1	Get DSS UR	EcDpPrEM	EcIoAdServer	Distributed Object	If the DSS UR for this Metadata Configuration File (MCF) is not already known in the PDPS database, EM searches the Advertiser for a "GetQueriableParameters" service for the desired MCF data type. This is accomplished via the IoAdApprovedSearchCommand class. Sinthe Advertiser is based on a replicated database, no specification is required to select the proper Advertiser. The local one is used.
C.13.1	Get Metadata Configuration File	EcDpPrEM	EcDsScienceDataServer	Distributed Object	DPS gets the metadata configuration file of the output data's ESDT (AST_09T). Datatype and version are from PDPS database; correct client name is from configuration file.
C.13.2	Run PGE	EcDpPrRunPGE	PGE<ACT>	command line	ACT is executed. Output files are placed in the output directory. The directory path is established by using a root, which was established by configuration, and the specific directory by the job id. This disk root is cross mounted by DPS, SDSRV and STMGT. This is to ensure that they are directly available to the DSS, for archival.
C.14.1	Get DSS UR	EcDpPrDM	EcIoAdServer	Distributed Object	If the DSS UR for this output data type is not already known in the PDPS database, DM searches the Advertiser for a "GetQueriableParameters" service for thesired output data type. This is accomplished via the IoAdApprovedSearchCommand class. Since the Advertiser is based on a replicated database, no specification is required to select the proper Advertiser. The local one is used.
C.15.1	Connect to SDSRV	EcDpPrDM	EcDsScienceDataServer	Distributed Object	PRONG begins a session with the SDSRV by connecting.

Table 3.5.5.3-1. Component Interaction Table: ASTER Scenario - Backward Chaining (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
C.15.2	Insert Data	EcDpPrDM	EcDsScienceDataServer	Distributed Object	PRONG requests that the newly created files for the AST_09T granule is inserted into the Data Server. An Insert request, containing the names of the files comprising the granule, is created for each granule. The structure of the Insert Request is hardcoded. SDSRV validates metadata and determines the archived names of the files.
C.15.3	STMGT Store	EcDsScienceDataServer	EcDsStArchiveServer	Distributed Object	SDSRV requests that the files are archived. The archive server must be able to read the inserted files directly from the DPS disks that they are residing on. The correct archive object to request is determined from collection level metadata for the ESDT, defined in the ESDT's descriptor.
C.15.4	Adding a Granule to Inventory	EcDsScienceDataServer	Sybase/SQS	CtLib	The validated metadata is parsed and added to the inventory of the SDSRV.
C.16.1	Trigger Event	EcDsScienceDataServer	EcSbSubServer	Distributed Object	Upon successful insertion of AST_09T the AST_09T:Insert event is triggered. The correct subscription server is determined from the SDSRV configuration. The correct events to trigger are determined from the events file, where they were stored when the ESDT was installed in the Data Server. Provided with the event triggering is the UR of the inserted granule.
C.16.2	Fire Subscriptions	EcSbSubServer	Sybase	CtLib	SBSRV queries the Sybase database determining which subscriptions need to be activated, or fired. Each query "hit" is an activated subscription and will execute independently.
C.17.1	Asynchronous Direct Notification	EcSbSubServer	EcPlSubMgr		The SBSRV will notify PLS that there is a new AST_09T granule available. The UR of the granule is passed in the notification to the user, along with a reference to the subscription that is being fulfilled. Direct Notification is to a Queuename (See Message Passing Key Mechansim) that PLS- Subscription Manager, provided when the subscription was submitted.
C.17.2	Connect to SDSRV	EcPlSubMgr	EcDsScienceDataServer	Distributed Object	Subscription Manager begins a session with the SDSRV by connecting, in order to determine the use of the new granule. The correct SDSRV is determined by using the GranuleUR in the notification message.
C.17.3	Add granules to Session	EcPlSubMgr	EcDsScienceDataServer	Distributed Object	Subscription Manager establishes the data context of the session with the SDSRV by adding AST_09 granule's UR. The GranuleUR of the AST_09 granule is added to the ESDTReferenceCollection.

Table 3.5.5.3-1. Component Interaction Table: ASTER Scenario - Backward Chaining (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
C.17.4	Retrieve Granule Metadata from Inventory	EcDsScienceDataServer	Sybase/SQS	CtLib	SDSRV completes establishing the data context by retrieving the metadata for the requested PGE granule from the Sybase/SQS database. The metadata for the PGE granule is passed back to the reference objects for each granule.
C.17.5	Inspect Granule Value Parameters	EcPlSubMgr	EcDsScienceDataServer	Distributed Object	Subscription Manager checks the new granule's metdata attributes (type, version, filesize and temporal range), to determine which, if any, jobs can use it as input.
C.18.1	Release Job	EcPlSubMgr	EcDpPrJobMgmt	rpc	Once it ensures that the input granule is to be used to run the job containing ETS from the PDPS database, Planning tells the Job Manager to release the job containing ETS, using the appropriate input granules.
C.18.2	Force Start Job	EcDpPrJobMgmt	event_demon	rpc	Job containing ETS is released.
C.18.3	Job Processing	event_demon	EcDpPrEM	command line	The job containing the ETS begins processing.
C.18.4	Connect to SDSRV	EcDpPrEM	EcDsScienceDataServer	Distributed Object	Processing begins a session with the SDSRV by connecting, in order to acquire the ETS PGE. The correct SDSRV is determined by using the GranuleUR of the PGE granule, which is defined in the Production plan and is part of the DPR.
C.18.5	Add granules to Session	EcDpPrEM	EcDsScienceDataServer	Distributed Object	PRONG establishes the data context of the session with the SDSRV by adding PGE granule's UR. The GranuleUR of the PGE granule is added to the ESDTReferenceCollection.
C.18.6	Retrieve Granule Metadata from Inventory	EcDsScienceDataServer	Sybase/SQS	CtLib	SDSRV completes establishing the data context by retrieving the metadata for the requested PGE granule from the Sybase/SQS database. The metadata for the PGE granule is passed back to the reference objects for each granule.
C.18.7	Acquire Data	EcDpPrEM	EcDsScienceDataServer	Distributed Object	DPS requests granules by submitting an Acquire request for the PGE granule. The Acquire request is for an ftpPush of all granules in the ESDTReferenceCollection. This request is synchronous, meaning that the return of the submit call of the request will contain the results of the request. This means that the response will not be sent until the PGE granule files have been ftp'd to the DPS disks. This request asks for no distribution notice to be email'd. The Acquire request structure is hardcoded.

Table 3.5.5.3-1. Component Interaction Table: ASTER Scenario - Backward Chaining (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
C.18.8	Create Staging Disk	EcDsScienceDataServer	EcStStagingDiskServer	Distributed Object	SDSRV creates Staging Disk for the metadata file, which allocates space and passes back a reference to that disk space. The reference to the Staging Disk is determined from the ArchiveID metadata of the granule to be distributed. The amount of staging disk to request is a default, hardcoded as a SDSRV constant.
C.18.9	Create Metadata file	EcDsScienceDataServer			The SDSRV creates a file containing the PGE granule's metadata before passing to Distribution.
C.18.10	Distribute Granules, Synchronous	EcDsScienceDataServer	EcDsDdRequestMgrMain	Distributed Object	SDSRV submits a request to DataDistribution. The request includes, for each granule, a reference to the metadata file as well as all data files. Other parameters from the Acquire request are passed to DDIST.
C.18.11	Create Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST creates Staging Disk for the granule files in the archive. This allocates space and passes back a reference to that disk space. The reference to the Staging Disk is determined from the information passed by the SDSRV in the distribution request ,which was the ArchiveID metadata parameter of the granule to be staged. The amount of staging disk to request is calculated from the file sizes of archived files, in the information passed in the Distribution Request.
C.18.12	STMGT Retrieve	EcDsDdRequestMgrMain	EcDsStArchiveServer	Distributed Object	DDIST requests that STMGT retrieve the PGE granule file that is archived. This will result in the file being staged to read-only cache disks. This means that all files needed to fulfill the distribution request are on disk, and ready to be copied. The correct archive object to request is determined from the information provided by the SDSRV in the distribution request. This returns references to the files in the read-only cache.
C.18.13	Link files to Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST links the files from the read-only cache into the staging disk.
C.18.14	Copy files to Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST copies the metadata files from the SDSRV's Staging Disk into the staging disk.
C.18.15	ftpPush Files	EcDsDdRequestMgrMain	EcDsStFtpDisServer	Distributed Object	DDIST now creates the Resource manager for ftp Pushes via a Resource Manager Factory. The correct resource manager is determined from the MediaType handed to the resource factory (ftpPush, in this case). The correct FtpServer is determined from configuration within the resource factory. The files, host, location, username and password are all determined from the information provided in the original Acquire request.

Table 3.5.5.3-1. Component Interaction Table: ASTER Scenario - Backward Chaining (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
C.18.16	ftp Files	EcDsStFtpDisServer	EcDpPrEM	rpc	CSS performs the actual low level ftp of the PGE files.
C.19.1	Connect to SDSRV	EcDpPrDM	EcDsScienceDataServer	Distributed Object	PRONG begins a session with the SDSRV by connecting. The correct SDSRV is determined by using the GranuleUR of the input granule.
C.19.2	Add granules to Session	EcDpPrDM	EcDsScienceDataServer	Distributed Object	PRONG establishes the data context of the session with the SDSRV by adding the input granule (1 AST_ANC) to the session. The GranuleUR of the input granule is added to the ESDTReferenceCollection. Note that this sequence is performed for each input granule, one at a time.
C.19.3	Retrieve Granule Metadata from Inventory	EcDsScienceDataServer	Sybase/SQS	CtLib	SDSRV completes establishing the data context by retrieving the metadata for the requested granule from the Sybase/SQS database. The metadata for each granule is passed back to the reference objects for each granule.
C.19.4	Acquire Data	EcDpPrDM	EcDsScienceDataServer	Distributed Object	DPS requests granules by submitting an Acquire request for those granules. The Acquire request is for an ftpPush of all granules in the ESDTReferenceCollection. This request is synchronous, meaning that the return of the submit call of the request will contain the results of the request. This means that the response will not be sent until the granule files have been ftp'd to the DPS disks. This request asks for no distribution notice to be email'd. The Acquire request structure is hardcoded.
C.19.5	Create Staging Disk	EcDsScienceDataServer	EcStStagingDiskServer	Distributed Object	SDSRV creates Staging Disk for metadata files, which allocates space and passes back a reference to that disk space. The reference to the Staging Disk is determined from the ArchiveID metadata of the granule to be distributed. The amount of staging disk to request is determined from collection level metadata from the ESDT's Descriptor file.
C.19.6	Create Metadata file	EcDsScienceDataServer			For each granule referenced in the Acquire request, the SDSRV creates a file containing the granule's metadata before passing to Distribution.
C.19.7	Distribute Granules, Synchronous	EcDsScienceDataServer	EcDsDdRequestMgrMain	Distributed Object	SDSRV submits a request to DataDistribution. The request includes, for each granule, a reference to the metadata file as well as all data files. Other parameters from the Acquire request are passed to DDIST.

Table 3.5.5.3-1. Component Interaction Table: ASTER Scenario - Backward Chaining (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
C.19.8	Create Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST creates Staging Disk for the granule files in the archive. This allocates space and passes back a reference to that disk space. The reference to the Staging Disk is determined from the information passed by the SDSRV in the distribution request ,which was the ArchiveID metadata parameter of the granule to be staged. The amount of staging disk to request is calculated from the file sizes in the information passed in the Distribution Request.
C.19.9	STMGT Retrieve	EcDsDdRequestMgrMain	EcDsStArchiveServer	Distributed Object	DDIST requests that STMGT retrieve the granule file that is archived. This will result in the file being staged to read-only cache disks. This means that all files needed to fulfill the distribution request are on disk, and ready to be copied. The correct archive object to request is determined from the information provided by the SDSRV in the distribution request. This returns references to the files in the read-only cache.
C.19.10	Link files to Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST links the files from the read-only cache into the staging disk.
C.19.11	Copy files to Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST copies the metadata files from the SDSRV's Staging Disk into the staging disk.
C.19.12	ftpPush Files	EcDsDdRequestMgrMain	EcDsStFtpDisServer	Distributed Object	DDIST now creates the Resource manager for ftp Pushes via a Resource Manager Factory. The correct resource manager is determined from the MediaType handed to the resource factory (ftpPush, in this case). The correct FtpServer is determined from configuration within the resource factory. The files, host, location, username and password are all determined from the information provided in the original Acquire request.
C.19.13	ftp Files	EcDsStFtpDisServer	EcDpPrDM	rpc	CSS performs the actual low level ftp of the files.
C.20.1	Get DSS UR	EcDpPrEM	EcIoAdServer	Distributed Object	If the DSS UR for this Metadata Configuration File (MCF) is not already known in the PDPS database, EM searches the Advertiser for a "GetQueriableParameters" service for the desired MCF data type. This is accomplished via the IoAdApprovedSearchCommand class. Sinthe Advertiser is based on a replicated database, no specification is required to select the proper Advertiser. The local one is used.

Table 3.5.5.3-1. Component Interaction Table: ASTER Scenario - Backward Chaining (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
C.21.1	Get Metadata Configuration File	EcDpPrEM	EcDsScienceDataServer	Distributed Object	DPS gets the metadata configuration file of the output data's ESDT (AST_08 and AST_05). Datatype and version are from PDPS database; correct client name is from configuration file.
C.21.2	Run PGE	EcDpPrRunPGE	PGE<ETS>	command line	ETS is executed. Output files are placed in the output directory. The directory path is established by using a root, which was established by configuration, and the specific directory by the job id. This disk root is cross mounted by DPS, SDSRV and STMGT. This is to ensure that they are directly available to the DSS, for archival.
C.22.1	Get DSS UR	EcDpPrDM	EcIoAdServer	Distributed Object	If the DSS UR for this output data type is not already known in the PDPS database, DM searches the Advertiser for a "GetQueriableParameters" service for thesired output data type. This is accomplished via the IoAdApprovedSearchCommand class. Since the Advertiser is based on a replicated database, no specification is required to select the proper Advertiser. The local one is used.
C.23.1	Connect to SDSRV	EcDpPrDM	EcDsScienceDataServer	Distributed Object	PRONG begins a session with the SDSRV by connecting.
C.23.2	Insert Data	EcDpPrDM	EcDsScienceDataServer	Distributed Object	PRONG requests that the newly created files for the AST_08 and AST_05 granules are inserted into the Data Server. An Insert request, containing the names of the files comprising the granule, is created for each granule. The structure of the Insert Request is hardcoded. SDSRV validates metadata and determines the archived names of the files. Note that these inserts occur one granule at a time.
C.23.3	STMGT Store	EcDsScienceDataServer	EcDsStArchiveServer	Distributed Object	SDSRV requests that the files are archived. The archive server must be able to read the inserted files directly from the DPS disks that they are residing on. The correct archive object to request is determined from collection level metadata for the ESDT, defined in the ESDT's descriptor.
C.23.4	Adding a Granule to Inventory	EcDsScienceDataServer	Sybase/SQS	CtLib	The validated metadata is parsed and added to the inventory of the SDSRV.

Table 3.5.5.3-1. Component Interaction Table: ASTER Scenario - Backward Chaining (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
C.24.1	Trigger Event	EcDsScienceDataServer	EcSbSubServer	Distributed Object	Upon successful insertion of AST_08 the AST_08:Insert event is triggered. The correct subscription server is determined from the SDSRV configuration. The correct events to trigger are determined from the events file, where they were stored when the ESDT was installed in the Data Server. Provided with the event triggering is the UR of the inserted granule.
C.24.2	Fire Subscriptions	EcSbSubServer	Sybase	CtLib	SBSRV queries the Sybase database determining which subscriptions need to be activated, or fired. Each query “hit” is an activated subscription and will execute independently.
C.25.1	Send E-mail	EcSbSubServer	Science User		The SBSRV builds an email notification that the user’s subscription on the AST_08:Insert event has been fired. This notification will identify the event, the subscription ID, the GranuleUR that was inserted and the previously supplied UserString.
C.26.1	Connect to SDSRV	EcSbSubServer	EcDsScienceDataServer	Distributed Object	In order to fulfill a standing order, the SBSRV begins a session with the SDSRV, on behalf of the subscription user. The correct SDSRV is determined by the Granule UR provided with the event triggering.
C.26.2	Add granules to Session	EcSbSubServer	EcDsScienceDataServer	Distributed Object	The SBSRV establishes the data context of the session with the SDSRV by adding the input granules to the session. The GranuleUR of each input granule is added to the ESDTReferenceCollection.
C.26.3	Retrieve Granule Metadata from Inventory	EcDsScienceDataServer	Sybase/SQS	CtLib	SDSRV completes establishing the data context by retrieving the metadata for the requested granules from the Sybase/SQS database. The metadata for each granule is passed back to the reference objects for each granule.
C.26.4	Acquire Data	EcSbSubServer	EcDsScienceDataServer	Distributed Object	SBSRV fulfills the standing order for the AST_08 granule by submitting an Acquire request for the granule. The Acquire request is for an ftpPush of all granules in the ESDTReferenceCollection. This request is asynchronous, meaning that the return of the submit call of the request will only contain the status of the request’s submittal. This request asks for a distribution notice to be email’d to the client. The Acquire request structure was hardcoded within the subscription server.

Table 3.5.5.3-1. Component Interaction Table: ASTER Scenario - Backward Chaining (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
C.26.5	Create Staging Disk	EcDsScienceDataServer	EcStStagingDiskServer	Distributed Object	SDSRV creates Staging Disk for metadata files, which allocates space and passes back a reference to that disk space. The reference to the Staging Disk is determined from the ArchiveID metadata of the granule to be distributed The amount of staging disk to request is determined from collection level metadata from the ESDT's Descriptor file.
C.26.6	Create Metadata file	EcDsScienceDataServer			For each granule referenced in the Acquire request, the SDSRV creates a file containing the granule's metadata before passing to Distribution.
C.26.7	Distribute Granules, Synchronous	EcDsScienceDataServer	EcDsDdRequestMgrMain	Distributed Object	SDSRV submits a request to Data Distribution. The request includes, for the granule, a reference to the metadata file as well as the data file. Other parameters from the Acquire request are passed to DDIST.
C.26.8	Create Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST creates Staging Disk for the granule files in the archive. This allocates space and passes back a reference to that disk space. The reference to the Staging Disk is determined from the information passed by the SDSRV in the distribution request ,which was the ArchiveID metadata parameter of the granule to be staged. The amount of staging disk to request is calculated from the file sizes in the information passed in the Distribution Request.
C.26.9	STMGT Retrieve	EcDsDdRequestMgrMain	EcDsStArchiveServer	Distributed Object	DDIST requests that STMGT retrieve the granule file that is archived. This will result in the file being staged to read-only cache disks. This means that all files needed to fulfill the distribution request are on disk, and ready to be copied. The correct archive object to request is determined from the information provided by the SDSRV in the distribution request. This returns references to the files in the read-only cache.
C.26.10	Link files to Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST links the files from the read-only cache into the staging disk.
C.26.11	Copy files to Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST copies the metadata files from the SDSRV's Staging Disk into the staging disk.

Table 3.5.5.3-1. Component Interaction Table: ASTER Scenario - Backward Chaining (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
C.26.12	ftpPush Files	EcDsDdRequestMgrMain	EcDsStFtpDisServer	Distributed Object	DDIST now creates the Resource manager for ftp Pushes via a Resource Manager Factory. The correct resource manager is determined from the MediaType handed to the resource factory (ftpPush, in this case). The correct FtpServer is determined from configuration within the resource factory. The files, host, location, username and password are all determined from the information provided in the original Acquire request.
C.27.1	ftp Files	EcDsStFtpDisServer	MODIS IT	rpc	CSS performs the actual low level ftp of the files.
C.28.1	Build Distribution Notice	EcDsDdRequestMgrMain			The DDIST builds an email notification that the user's order has been fulfilled. This notification will include the media id, type and format, as well as UR, type and file names and sizes for each granule.
C.28.2	Send E-mail	EcDsDdRequestMgrMain	MODIS IT		DDIST sends the distribution notice to the user via email. Since Standing Orders are not currently tracked orders, and the user DDIST sends the notification to is determined from the Order, the notice is currently sent to a pre-configured default Email address, for DAAC Distribution Technician parsing and forwarding.

3.5.6 QA Metadata Update Thread

3.5.6.1 Interaction Diagram - Domain View

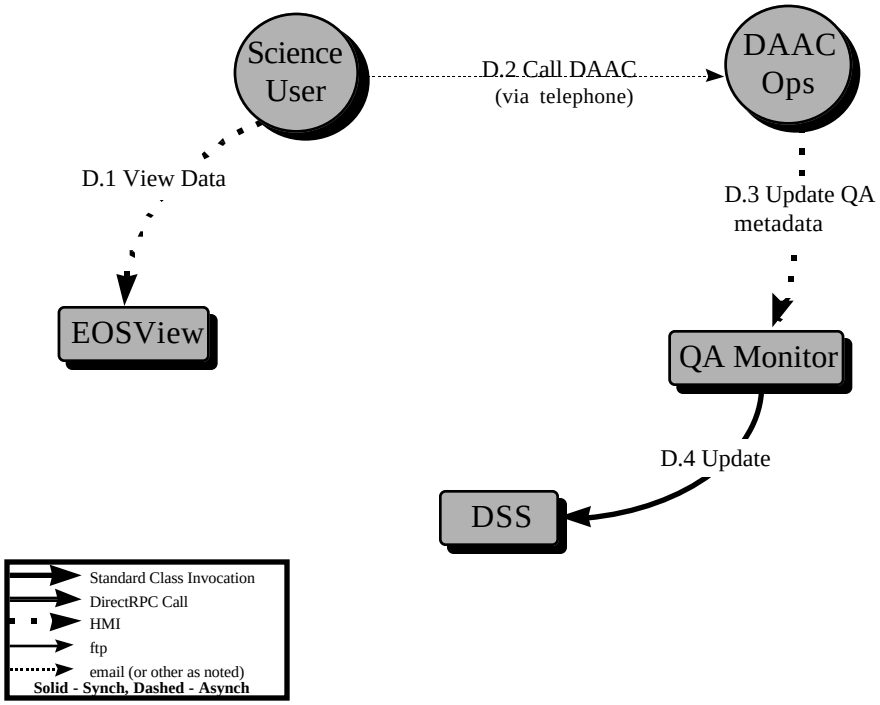

Figure 3.5.6.1-1. QA Metadata Update Interaction Diagram

3.5.6.2 Interaction Table - Domain View

Table 3.5.6.2-1. Interaction Table - Domain View: ASTER Scenario, QA Metadata Update

Step	Interaction	Interface Client	Interface Provider	Data Issues	Preconditions	Description
D.1	View Data	Science User	EOSView			Upon notification that theAST_08 has been placed on their workstation, the Scientist views the AST_08 data with EOSView.
D.2	Call DAAC	Science User	DAAC Science Data Specialist			The Scientist QA's the produced data. S/he calls the DAAC, informing the DAAC Science Data Specialist that the granule's QA flags should be updated.
D.3	Update QA Metadata	DAAC Science Data Specialist	QAMonitor			DAAC Science Data Specialist uses the QAMonitor tool to update the Science QA metadata of the granule.
D.4	Update	QA Monitor	DSS			QAMonitor invokes the Update service offered by the Data Server on the granule. The QAMonitor passes the Scientists requested QA values to the DSS for permanent updating of the granule's metadata.

3.5.6.3 Component Interaction Table

Table 3.5.6.3-1. Component Interaction Table: ASTER Scenario - QA Metadata Update

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
D.1.1	Invoke EOSView	Science User	EOSView	Xevent	Science User begins the EOSView application. While logged on to the Science Desktop, the user double clicks the EOSView icon.
D.1.2	Display AST_08 Data	Science User	EOSView	GUI	The Science User specifies which file to display and sets visualization parameters. The data file is now displayed for the user.
D.3.1	Invoke DAAC QA Monitor	DAAC Science Data Specialist	EcDpPrQa MonitorGUI	Xevent	DAAC Science Data Specialist begins the QA Monitor application. While logged on to the DAAC Desktop, the user double clicks the QA Monitor icon.
D.3.2	Establish QA values	DAAC Science Data Specialist	EcDpPrQa MonitorGUI	GUI	DAAC Science Data Specialist establishes the updated values for selected metadata fields, for the selected granules. Granules are select by selecting data type and temporal range. Fields to update are hardwired.
D.3.3	Connect to SDSRV	EcDpPrQa MonitorGUI	EcDsScien ceDataServ er	Distribute d Object	QA Monitor begins a session with the SDSRV by connecting, in order to find granules to be updated. The correct SDSRV is determined by using the ServerUR indicated in configuration, based on data type.
D.3.4	SDSRV Query	EcDpPrQa MonitorGUI	EcDsScien ceDataServ er	Distribute d Object	The Gateway builds a DsClQuery object. This object is handed to the Search interface of the DsClESDTReferenceCollector. This Search method is synchronous, so the results of the search will be returned to the calling function.
D.3.5	Querying the Inventory	EcDsScien ceDataServ er	Sybase/SQ S	CtLib	The SDSRV breaks down the Query object and translates it into a sequence of calls to the inventory database. Resultant rows are converted into data granules, each with their metadata extracted from the database. These results are packaged and returned to the Query client.
D.3.6	Inspect Granule Value Parameters	EcDpPrQa MonitorGUI	EcDsScien ceDataServ er	Distribute d Object	QA Monitor inspects each resultant granule for the values of displayed metadata fields.
D.3.7	Select granules to update	DAAC Science Data Specialist	EcDpPrQa MonitorGUI	GUI	DAAC Science Data Specialist selects granules for updating.
D.4.1	Update Granule metadata	EcDpPrQa MonitorGUI	EcDsScien ceDataServ er	Distribute d Object	QA Monitor submits an update request for the granules to be updated (one granule at a time). The structure of the Update request is hardcoded.
D.4.2	Update a metdata inventory	EcDsScien ceDataServ er	Sybase/SQ S	CtLib	SDSRV updates the metadata inventory attributes for the granules that are being updated.

3.5.7 On-Demand Production Thread

3.5.7.1 Interaction Diagram - Domain View

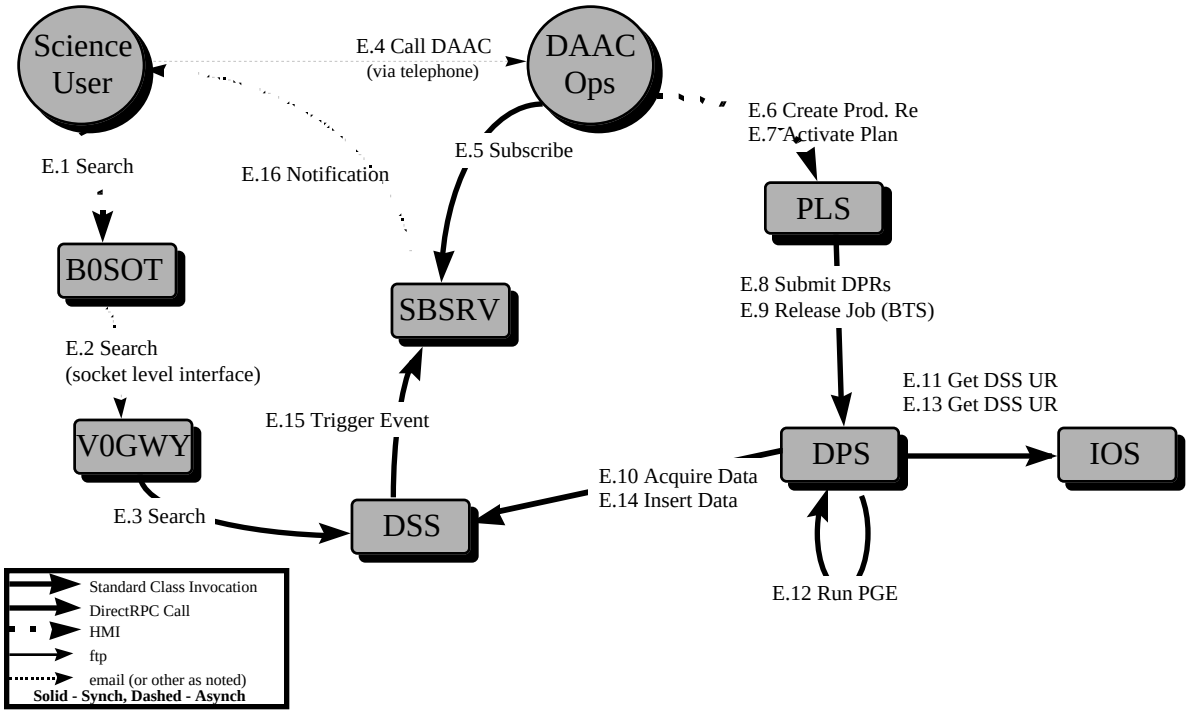

Figure 3.5.7.1-1. On-Demand Production Interaction Diagram

3.5.7.2 Interaction Table - Domain View

Table 3.5.7.2-1. Interaction Table - Domain View: ASTER Scenario, On-Demand Production (1 of 2)

Step	Interaction	Interface Client	Interface Provider	Data Issues	Preconditions	Description
E.1	Inventory Search	Science User	B0SOT			Scientist searches ECS holdings for ASTER images that are over his/her area of study.
E.2	Search	B0SOT	GTWAY			B0SOT submits the Science User's search criteria to the V0 Gateway in ODL format, via a well defined socket.
E.3	Search	GTWAY	SDSRV			The V0 gateway translates the Search criteria from ODL to a query object (using GlParameters), and submits that query to the Search service. The results of this Search are returned synchronously, and are passed back to B0SOT, which displays them to the DAAC Operator.
E.4	Call DAAC	Science User	DAAC Production Planner			The scientist discovers that one AST_L1Bt granule in particular is perfectly suited to support his/her science study. The scientist discovers that the granule can be transformed into a L2 Brightness Temperature (AST_04) product. The scientist calls the DAAC, asking for the granule to be processed to L2, and to be notified when that processing is complete.
E.5	Subscribe	DAAC Production Planner	SBSRV			The DAAC Production Planner places a subscription for the Science User to be notified when the AST_04 is available.
E.6	Create Production Request	DAAC Production Planner	PLS			The DAAC Production Planner creates a production request containing the requested BTS algorithm to be run.
E.7	Activate Production Plan	DAAC Production Planner	PLS		PGEs passed SSI&T. Plan already created.	DAAC Production Planner modifies and activates a plan, which includes a DPR for the BTS PGE to be run on the requested AST_L1BT.
E.8	Submit DPRs	PLS	DPS			DPR for BTS is submitted to DPS.
E.9	Release Job	PLS	DPS			Since all inputs are available to run the BTS PGE, references to those input granules are passed to DPS, and the BTS job is released.

Table 3.5.7.2-1. Interaction Table - Domain View: ASTER Scenario, On-Demand Production (2 of 2)

Step	Interaction	Interface Client	Interface Provider	Data Issues	Preconditions	Description
E.10	Acquire Data	DPS	DSS	1 AST_ANC @772MB, 1 AST_L1BT @5MB	AST_ANC data already inserted	DPS submits Acquire Request for input granules, via ftpPush, for input to BTS.
E.11	Get DSS UR	DPS	IOS			DPS gets the DSS UR from Advertiser.
E.12	Run PGE			1 AST_04 @4.6 MB produced		BTS runs, creating AST_04 granules.
E.13	Get DSS UR	DPS	IOS			DPS gets the DSS UR from Advertiser.
E.14	Insert Data	DPS	DSS		AST_04 ESDT	Archive newly created AST_04 granule.
E.15	Trigger Event	SDSRV	SBSRV			Trigger AST_04:Insert event.
E.16	Notification	SBSRV	Science User			Send email notification to Science User, notifying that the AST_04 granule has been produced. Notification message includes the UR of the AST_04 granule.

3.5.7.3 Component Interaction Table

Table 3.5.7.3-1. Component Interaction Table: ASTER Scenario - On-Demand Production

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
E.1.1	Startup B0SOT	Science User	xims	XEvent	Science User invokes the B0SOT GUI Application. The user has already been logged on the DAAC Desktop and begins B0SOT by double-clicking on an icon.
E.1.2	Select Inventory Search, Provide Query constraints, Submit Query	Science User	xims	GUI	The Science User provides search constraints and the products desired. When query constraints are completed, the query is submitted.

Table 3.5.7.3-1. Component Interaction Table: ASTER Scenario - On-Demand Production (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
E.2.1	V0 Gateway Inventory	xims	EcDmV0ToEcsGateway	ODL, over sockets	B0SOT submits a search to the V0 Gateway, by converting the search criteria into an ODL structure and passing that structure to a socket provided by the Gateway. The correct socket is determined from configuration information contained in the Valids file.
E.3.1	Establish ECS User	EcDmV0ToEcsGateway	MsAcManager	Distributed Object	V0 Gateway retrieves the User Profile using ECS Authenticator from ODL message, which includes an encrypted UserID and Password. The User Registration Server is replicated across DAACs, so the connection is made to the local User Registration Server.
E.3.2	Translate Terms	EcDmV0ToEcsGateway	EcDmDictServer	Ctlib (RWDbTool)	Gateway translates the V0 terms from ODL into ECS names for query submittal. Interface currently is directly to Data Dictionary database. Database name is retrieved from configuration file.
E.3.3	Connect to SDSRV	EcDmV0ToEcsGateway	EcDsScienceDataServer	Distributed Object	The Gateway first connects to the SDSRV. The correct SDSRV is determined from configuration information.
E.3.4	SDSRV Query	EcDmV0ToEcsGateway	EcDsScienceDataServer	Distributed Object	The Gateway translates the query into a DsClQuery object. This object is handed to the Search interface of the DsClESDTReferenceCollector. This Search method is synchronous, so the results of the search will be returned to the calling function.
E.3.5	Querying the Inventory	EcDsScienceDataServer	Sybase/SQS	CtLib	The SDSRV breaks down the Query object and translates it into a sequence of calls to the inventory database. Resultant rows are converted into data granules, each with their metadata extracted from the database. These results are packaged and returned to the Query client.
E.3.6	Result Retrieval	xims	EcDmV0ToEcsGateway	ODL, over Sockets	When the Gateway gets the results, they are translated into ODL, and passed back to the B0SOT tool. The correct socket for sending results to B0SOT is the one used to submit the query. B0SOT then displays the results of the query to the user.
E.5.1	Startup SBSRV GUI	DAAC User Services Representative	EcSbGui	Xterm	DAAC User Services Representative invokes SBSRV GUI application.

Table 3.5.7.3-1. Component Interaction Table: ASTER Scenario - On-Demand Production (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
E.5.2	Create & Submit Subscription from GUI	DAAC User Services Representative	EcSbGui	Xterm	DAAC User Services Representative represents him/herself as the Science User. The DAAC Operator brings up the GUI and clicks button to create new subscription. A list of events is then displayed from which the op can choose to subscribe. DAAC Operator selects the AST_04:Insert Event for subscription. Only one action (besides notification), is available from the SBSRV at this time. FtpPush as a distribution mechanism is input via a GUI button. Other parameters required for FtpPush, including the Science User's host name, target directory, ftp user name, and ftp password, are input via the GUI.
E.5.3	Submit Subscription	EcSbGui	EcSbSubServer	Distributed Object	Submit the subscription to the Subscription Server. This is accomplished with the EcClSubscription interface class. The correct SBSRV is determined via a ServerUR, declared in configuration.
E.5.4	Persist a Subscription	EcSbSubServer	Sybase	CtLib	Subscription is stored in the Sybase Database.
E.6.1	Startup Production Request Editor	DAAC Productrion Planner	EcPlPREditor_IF	GUI	DAAC Planner invokes the Production Request Editor. While already running within the DAAC Desktop, the planner double clicks on the Planning Workbench icon.
E.6.2	Build Production Requests	DAAC Production Planner	EcPlPREditor_IF	GUI	DAAC Planner creates Production Requests for the BTS algorithm. Algorithm (BTS) is selected, along with the time domain of the output (and input) data.
E.6.3	Search for Service Advertisements	EcPlPREditor_IF	EcIoAdServer	Distributed Object	In order to ensure that the correct input data is used for the BTS algorithm, the Editor searches Advertiser for the service the Subscribe to AST_L1BT:Insert event. This is accomplished via the IoAdApprovedSearchCommand class. Since the Advertiser is based on a replicated database, no specification is required to select the proper Advertiser. The local one is used. Note: This (and the following SBSRV steps) only occurs if Planning does not already have a subscription for the AST_L1BT:Insert event.
E.6.4	Connect to SBSRV	EcPlPREditor_IF	EcSbSubServer	Distributed Object	The Editor connects to the subscription server in order to subscribe for notification of new AST_L1BT granules. The correct Subscription server is determined from the Subscribe Advertisement.
E.6.5	Submit Subscription	EcPlPREditor_IF	EcSbSubServer	Distributed Object	Submit the subscription to the Subscription Server. This is accomplished with the EcClSubscription interface class.

Table 3.5.7.3-1. Component Interaction Table: ASTER Scenario - On-Demand Production (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
E.6.6	Persist a Subscription	EcSbSubServer	Sybase	CtLib	Subscription is stored in the Sybase Database.
E.6.7	Search for Service Advertisements	EcPlPREditor_IF	EcIoAdServer	Distributed Object	In order to determine where (which SDSRV) the input data (AST_L1BT) is located the Editor searches the Advertiser for a "GetQueriableParameters" service for the desired input data type. This is in lieu of searching for Product Advertisements. This is accomplished via the IoAdApprovedSearchCommand class. Since the Advertiser is based on a replicated database, no specification is required to select the proper Advertiser. The local one is used. Note: AST_ANC and GDAS_0ZF are considered "static" data, and have references stored after SSI&T.
E.6.8	Connect to SDSRV	EcPlPREditor_IF	EcDsScienceDataServer	Distributed Object	Looking for input granules for the BTS PGE, the Editor first connects to the SDSRV. The correct SDSRV is determined from the service provider on the GetQueriableParameters Advertisement.
E.6.9	SDSRV Query	EcPlWb	EcDsScienceDataServer	Distributed Object	The Workbench builds a DsClQuery object, looking for AST_L1BT granules that match space and time parameters. This object is handed to the Search interface of the DsClESDTReferenceCollector. This Search method is synchronous, so the results of the search will be returned to the calling function.
E.6.10	Querying the Inventory	EcDsScienceDataServer	Sybase/SQS	CtLib	The SDSRV breaks down the Query object and translates it into a sequence of calls to the inventory database. Resultant rows are converted into data granules, each with their metadata extracted from the database. These results are packaged and returned to the Query client.
E.6.11	Inspect Granule Value Parameters	EcPlPREditor_IF	EcDsScienceDataServer	Distributed Object	Editor checks the granule's metdata attributes (type, version, filesize and temporal range), to establish job dependencies. References to desired granules are packaged in the BTS DPR.
E.7.1	Startup Planning Workbench	DAAC Operator - Planner	EcPlWb	GUI	DAAC Planner invokes the Planning workbench. While already running within the DAAC Desktop, the planner double clicks on the Planning Workbench icon.
E.7.2	Create a Plan	DAAC Operator - Planner	EcPlWb	GUI	Planner interacts with Planning Workbench GUI to create a plan which contains a DPR for the BTS PGE.
E.7.3	Activate a Plan	EcPlWb	EcDpPrJobMgmt	rpc	Updated plan is activated, making it the current processing plan.
E.8.1	Submit DPRs	EcPlWb	event_demon	rpc (COTS)	The DPR (containing BTS PGE) in updated plan are submitted, to AutoSys.

Table 3.5.7.3-1. Component Interaction Table: ASTER Scenario - On-Demand Production (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
E.9.1	Release Job	EcPlWb	EcDpPrJobMgmt	rpc	Planning tells the Job Manager to release the job containing BTS, using the appropriate input granules.
E.9.2	Force Start Job	EcDpPrJobMgmt	event_demon	rpc	Job containing BTS is released.
E.9.3	Job Processing	event_demon	EcDpPrEM	command line	The job containing the BTS begins processing.
E.9.4	Connect to SDSRV	EcDpPrEM	EcDsScienceDataServer	Distributed Object	Processing begins a session with the SDSRV by connecting, in order to acquire the BTS PGE. The correct SDSRV is determined by using the GranuleUR of the PGE granule, which is defined in the Production plan and is part of the DPR.
E.9.5	Add granules to Session	EcDpPrEM	EcDsScienceDataServer	Distributed Object	PRONG establishes the data context of the session with the SDSRV by adding PGE granule's UR. The GranuleUR of the PGE granule is added to the ESDTReferenceCollection.
E.9.6	Retrieve Granule Metadata from Inventory	EcDsScienceDataServer	Sybase/SQS	CtLib	SDSRV completes establishing the data context by retrieving the metadata for the requested PGE granule from the Sybase/SQS database. The metadata for the PGE granule is passed back to the reference objects for each granule.
E.9.7	Acquire Data	EcDpPrEM	EcDsScienceDataServer	Distributed Object	DPS requests granules by submitting an Acquire request for the PGE granule. The Acquire request is for an ftpPush of all granules in the ESDTReferenceCollection. This request is synchronous, meaning that the return of the submit call of the request will contain the results of the request. This means that the response will not be sent until the PGE granule files have been ftp'd to the DPS disks. This request asks for no distribution notice to be email'd. The Acquire request structure is hardcoded.
E.9.8	Create Staging Disk	EcDsScienceDataServer	EcStStagingDiskServer	Distributed Object	SDSRV creates Staging Disk for the metadata file, which allocates space and passes back a reference to that disk space. The reference to the Staging Disk is determined from the ArchiveID metadata of the granule to be distributed. The amount of staging disk to request is a default, hardcoded as a SDSRV constant.
E.9.9	Create Metadata file	EcDsScienceDataServer			The SDSRV creates a file containing the PGE granule's metadata before passing to Distribution.

Table 3.5.7.3-1. Component Interaction Table: ASTER Scenario - On-Demand Production (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
E.9.10	Distribute Granules, Synchronous	EcDsScienceDataServer	EcDsDdRequestMgrMain	Distributed Object	SDSRV submits a request to DataDistribution. The request includes, for each granule, a reference to the metadata file as well as all data files. Other parameters from the Acquire request are passed to DDIST.
E.9.11	Create Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST creates Staging Disk for the granule files in the archive. This allocates space and passes back a reference to that disk space. The reference to the Staging Disk is determined from the information passed by the SDSRV in the distribution request ,which was the ArchiveID metadata parameter of the granule to be staged. The amount of staging disk to request is calculated from the file sizes of archived files, in the information passed in the Distribution Request.
E.9.12	STMGT Retrieve	EcDsDdRequestMgrMain	EcDsStArchiveServer	Distributed Object	DDIST requests that STMGT retrieve the PGE granule file that is archived. This will result in the file being staged to read-only cache disks. This means that all files needed to fulfill the distribution request are on disk, and ready to be copied. The correct archive object to request is determined from the information provided by the SDSRV in the distribution request. This returns references to the files in the read-only cache.
E.9.13	Link files to Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST links the files from the read-only cache into the staging disk.
E.9.14	Copy files to Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST copies the metadata files from the SDSRV's Staging Disk into the staging disk.
E.9.15	ftpPush Files	EcDsDdRequestMgrMain	EcDsStFtpDisServer	Distributed Object	DDIST now creates the Resource manager for ftp Pushes via a Resource Manager Factory. The correct resource manager is determined from the MediaType handed to the resource factory (ftpPush, in this case). The correct FtpServer is determined from configuration within the resource factory. The files, host, location, username and password are all determined from the information provided in the original Acquire request.
E.9.16	ftp Files	EcDsStFtpDisServer	EcDpPrEM	rpc	CSS performs the actual low level ftp of the PGE files.
E.10.1	Connect to SDSRV	EcDpPrDM	EcDsScienceDataServer	Distributed Object	PRONG begins a session with the SDSRV by connecting. The correct SDSRV is determined by using the GranuleUR of the input granule.

Table 3.5.7.3-1. Component Interaction Table: ASTER Scenario - On-Demand Production (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
E.10.2	Add granules to Session	EcDpPrDM	EcDsScience DataServer	Distributed Object	PRONG establishes the data context of the session with the SDSRV by adding the input granules (1 AST_L1BT and 1 AST_ANC) to the session. The GranuleUR of the input granule is added to the ESDTReferenceCollection. Note that this sequence is performed for each input granule, one at a time.
E.10.3	Retrieve Granule Metadata from Inventory	EcDsScience DataServer	Sybase/SQS	CtLib	SDSRV completes establishing the data context by retrieving the metadata for the requested granule from the Sybase/SQS database. The metadata for each granule is passed back to the reference objects for each granule.
E.10.4	Acquire Data	EcDpPrDM	EcDsScience DataServer	Distributed Object	DPS requests granules by submitting an Acquire request for those granules. The Acquire request is for an ftpPush of all granules in the ESDTReferenceCollection. This request is synchronous, meaning that the return of the submit call of the request will contain the results of the request. This means that the response will not be sent until the granule files have been ftp'd to the DPS disks. This request asks for no distribution notice to be email'd. The Acquire request structure is hardcoded.
E.10.5	Create Staging Disk	EcDsScience DataServer	EcStStaging DiskServer	Distributed Object	SDSRV creates Staging Disk for metadata files, which allocates space and passes back a reference to that disk space. The reference to the Staging Disk is determined from the ArchiveID metadata of the granule to be distributed. The amount of staging disk to request is determined from collection level metadata from the ESDT's Descriptor file.
E.10.6	Create Metadata file	EcDsScience DataServer			For each granule referenced in the Acquire request, the SDSRV creates a file containing the granule's metadata before passing to Distribution.
E.10.7	Distribute Granules, Synchronous	EcDsScience DataServer	EcDsDdRequestMgrMain	Distributed Object	SDSRV submits a request to DataDistribution. The request includes, for each granule, a reference to the metadata file as well as all data files. Other parameters from the Acquire request are passed to DDIST.

Table 3.5.7.3-1. Component Interaction Table: ASTER Scenario - On-Demand Production (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
E.10.8	Create Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST creates Staging Disk for the granule files in the archive. This allocates space and passes back a reference to that disk space. The reference to the Staging Disk is determined from the information passed by the SDSRV in the distribution request ,which was the ArchiveID metadata parameter of the granule to be staged. The amount of staging disk to request is calculated from the file sizes in the information passed in the Distribution Request.
E.10.9	STMGT Retrieve	EcDsDdRequestMgrMain	EcDsStArchiveServer	Distributed Object	DDIST requests that STMGT retrieve the granule file that is archived. This will result in the file being staged to read-only cache disks. This means that all files needed to fulfill the distribution request are on disk, and ready to be copied. The correct archive object to request is determined from the information provided by the SDSRV in the distribution request. This returns references to the files in the read-only cache.
E.10.10	Link files to Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST links the files from the read-only cache into the staging disk.
E.10.11	Copy files to Staging Disk	EcDsDdRequestMgrMain	EcStStagingDiskServer	Distributed Object	DDIST copies the metadata files from the SDSRV's Staging Disk into the staging disk.
E.10.12	ftpPush Files	EcDsDdRequestMgrMain	EcDsStFtpDisServer	Distributed Object	DDIST now creates the Resource manager for ftp Pushes via a Resource Manager Factory. The correct resource manager is determined from the MediaType handed to the resource factory (ftpPush, in this case). The correct FtpServer is determined from configuration within the resource factory. The files, host, location, username and password are all determined from the information provided in the original Acquire request.
E.10.13	ftp Files	EcDsStFtpDisServer	EcDpPrDM	rpc	CSS performs the actual low level ftp of the files.
E.11.1	Get DSS UR	EcDpPrEM	EcIoAdServer	Distributed Object	If the DSS UR for this Metadata Configuration File (MCF) is not already known in the PDPS database, EM searches the Advertiser for a "GetQueriableParameters" service for the desired MCF data type. This is accomplished via the IoAdApprovedSearchCommand class. Sinthe Advertiser is based on a replicated database, no specification is required to select the proper Advertiser. The local one is used.

Table 3.5.7.3-1. Component Interaction Table: ASTER Scenario - On-Demand Production (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
E.12.1	Get Metadata Configuration File	EcDpPrEM	EcDsScience DataServer	Distributed Object	DPS gets the metadata configuration file of the output data's ESDT (AST_04). Datatype and version are from PDPS database, correct client name is from configuration file.
E.12.2	Run PGE	EcDpPrRunPGE	PGE<BTS>	command line	BTS is executed. Output files are placed in the output directory. The directory path is established by using a root, which was established by configuration, and the specific directory by the job id. This disk root is cross mounted by DPS, SDSRV and STMGT. This is to ensure that they are directly available to the DSS, for archival.
E.13.1	Get DSS UR	EcDpPrDM	EcIoAdServer	Distributed Object	If the DSS UR for this output data type is not already known in the PDPS database, DM searches the Advertiser for a "GetQueriableParameters" service for thesired output data type. This is accomplished via the IoAdApprovedSearchCommand class. Since the Advertiser is based on a replicated database, no specification is required to select the proper Advertiser. The local one is used.
E.12.1	Connect to SDSRV	EcDpPrDM	EcDsScience DataServer	Distributed Object	PRONG begins a session with the SDSRV by connecting.
E.12.2	Insert Data	EcDpPrDM	EcDsScience DataServer	Distributed Object	PRONG requests that the newly created files for the AST_04 granule are inserted into the Data Server. An Insert request, containing the names of the files comprising the granule, is created for each granule. The structure of the Insert Request is hardcoded. SDSRV validates metadata and determines the archived names of the files. Note that these inserts occur one granule at a time.
E.12.3	STMGT Store	EcDsScience DataServer	EcDsStArchiveServer	Distributed Object	SDSRV requests that the files are archived. The archive server must be able to read the inserted files directly from the DPS disks that they are residing on. The correct archive object to request is determined from collection level metadata for the ESDT, defined in the ESDT's descriptor.
E.12.4	Adding a Granule to Inventory	EcDsScience DataServer	Sybase/SQS	CtLib	The validated metadata is parsed and added to the inventory of the SDSRV.

Table 3.5.7.3-1. Component Interaction Table: ASTER Scenario - On-Demand Production (Continued)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
E.13.1	Trigger Event	EcDsScienceDataServer	EcSbSubServer	Distributed Object	Upon successful insertion of AST_04 the AST_04:Insert event is triggered. The correct subscription server is determined from the SDSRV configuration. The correct events to trigger are determined from the events file, where they were stored when the ESDT was installed in the Data Server. Provided with the event triggering is the UR of the inserted granule.
E.13.2	Fire Subscriptions	EcSbSubServer	Sybase	CtLib	SBSRV queries the Sybase database determining which subscriptions need to be activated, or fired. Each query “hit” is an activated subscription and will execute independently.
E.14.1	Send E-mail	EcSbSubServer	Science User		The SBSRV will notify the Science User that the AST_04 granule is available. The UR of the granule is passed in the notification to the user, along with a reference to the subscription that is being fulfilled.

3.5.8 Simplified Expedited Data Support Thread

3.5.8.1 Interaction Diagram - Domain View

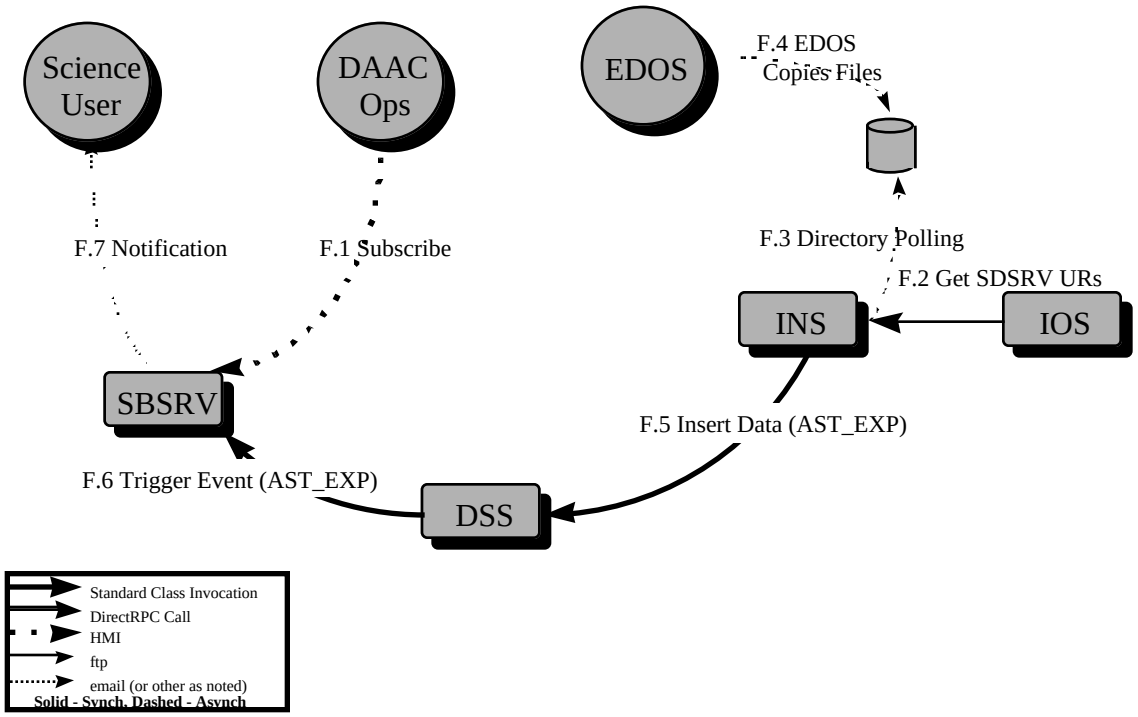

Figure 3.5.8.1-1. Simplified Expedited Data Support Interaction Diagram

3.5.8.2 Interaction Table - Domain View

Table 3.5.8.2-1. Interaction Table - Domain View: ASTER Scenario, Simplified Expedited Data

Step	Interaction	Interface Client	Interface Provider	Data Issues	Preconditions	Description
F.1	Subscribe	DAAC User Services Representative	SBSRV			The DAAC User Services Representative places a subscription for the Science User to be notified when the AST_EXP is available.
	Get SDSRV URs	INS	IOS			Upon startup, Ingest gets the SDSRV URs for each data type in its database.
F.2	Polling	INS	directory		Entire step is really a precondition.	When system is started, Ingest begins polling a directory, looking for files that meet the following standard: *.EDR.XFR, in the preconfigured directory.
F.3	Copy Files	EDOS	directory		EDOS knows the host and directory to place files.	EDOS copies the Expedited Data and metadata files to the directory which Ingest is polling.
F.4	Insert Data	INS	DSS	1 AST_EXP @ 16.6MB	AST_EXP ESDT	Ingest inserts the new ASTER Expedited granule into the Data Server.
F.5	Trigger Event	DSS	SBSRV			Upon successful completion of insertion of ASTER Expedited Data, the AST_EXP:Insert event is triggered.
F.6	Notification	SBSRV	Science User			Science User is notified, via email, that new ASTER Expedited Data is available. The notification contains the UR of the new AST_EXP granule.

3.5.8.3 Component Interaction Table

Table 3.5.8.3-1. Component Interaction Table: ASTER Scenario - Simplified Expedited Data (1 of 2)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
F.1.1	Startup SBSRV GUI	DAAC User Services Representative	EcSbGui	Xterm	DAAC User Services Representative invokes SBSRV GUI application.
F.1.2	Create & Submit Subscription from GUI	DAAC User Services Representative	EcSbGui	Xterm	DAAC User Services Representative represents him/herself as the Science User. The DAAC Operator brings up the GUI and clicks button to create new subscription. A list of events is then displayed from which the op can choose to subscribe. DAAC Operator selects the AST_EXP:Insert Event for subscription. Only one action (besides notification), is available from the SBSRV at this time. FtpPush as a distribution mechanism is input via a GUI button. Other parameters required for FtpPush, including the Science User's host name, target directory, ftp user name, and ftp password, are input via the GUI.
F.1.3	Submit Subscription	EcSbGui	EcSbSubServer	Distributed Object	Submit a subscription with ftp action to the Subscription Server. This is accomplished with the EcClSubscription interface class. The correct SBSRV is determined via a ServerUR, declared in configuration.
F.1.4	Persist a Subscription	EcSbSubServer	Sybase	CtLib	Subscription is stored in the Sybase Database.
	Get SDSRV URs from IOS	EcInReqMgr	EcIoAdServer	Distributed Object	Upon startup, Ingest Request Manager requests the SDSRV URs for each data type in its database.
F.2.1	Ingest Polling	EcInPolling			Ingest begins polling the configured directory. It will periodically look for files named *.EDR.XFR. The polling periodicity is determined from a file parameter configuration. The mask of the file to look for is determined from configuration.
F.3.1	EDOS Copies Files	EDOS		ftp	EDOS ftp's the ASTER Expedited Data to the predetermined directory. Location, directory, username and password are as per the ASTER-ECS ICD.
F.4.1	Polling Detects Files	EcInPolling			Ingest Polling detects files matching the *.EDR.XFR masks.
F.4.2	Ingest Request	EcInPolling	EcInReqMgr	Distributed Object	Auto Ingest process packages the file names into the appropriate Ingest Requests. The data source (EDOS), defined on startup, is passed to the Ingest Request Manager.

Table 3.5.8.3-1. Component Interaction Table: ASTER Scenario - Simplified Expedited Data (2 of 2)

Step	Event	Interface Client	Interface Provider	Interface Mech.	Description
F.2.3		EcInReqMgr	EcInGran	Distributed Object	Ingest Request Manager packages the request into granules and sends them to the Ingest Granule Server.
F.4.4	Connect to SDSRV	EcInGran	EcDsScienceDataServer	Distributed Object	Upon detecting an ASTER Expedited data file, Ingest begins a session with the SDSRV by connecting. The correct SDSRV is determined during EcInReqMgr startup, from Advertising, based on the data type.
F.4.5	Get Metadata Configuration File	EcInGran	EcDsScienceDataServer	Distributed Object	Ingest requests the metadata configuration file (mcf) for the data being inserted. The data types being inserted are derived from the Ingest Request messages sent by the Polling server.
F.4.6	Validate Metadata	EcInGran	EcDsScienceDataServer	Distributed Object	After building a metadata file for the CPF granule, Ingest asks SDSRV to validate the metadata, based on the granule's data type.
F.4.7	Insert Data	EcInGran	EcDsScienceDataServer	Distributed Object	Ingest requests that the received files for the AST_EXP are inserted into the Data Server. An Insert request, containing the names of the files comprising the Expedited Data granule, is created. The structure of the Insert Request is hardcoded in the granule server process. SDSRV re-validates metadata and determines the archived names of the files.
F.4.8	STMGT Store	EcDsScienceDataServer	EcDsStArchiveServer	Distributed Object	SDSRV requests that the Expedited Data is archived. The archive server reads the inserted files directly from the Ingest staging disks that they are residing on. The correct archive object to request is determined from collection level metadata for the ESDT, defined in the ESDT's descriptor.
F.4.9	Adding a Granule to Inventory	EcDsScienceDataServer	Sybase/SQS	CtLib	The validated metadata is parsed and added to the inventory of the SDSRV.
F.5.1	Trigger Event	EcDsScienceDataServer	EcSbSubServer	Distributed Object	Upon successful insertion of AST_EXP granule, the AST_EXP:Insert event is triggered. The correct subscription server is determined from SDSRV configuration. The correct events to trigger are determined from the events file, which was populated during ESDT installation. Provided with the event triggering is the UR of the inserted granule.
F.5.2	Fire Subscriptions	EcSbSubServer	Sybase	CtLib	SBSRV queries the Sybase database determining which subscriptions need to be activated, or fired. Each query "hit" is an activated subscription and will execute independently.
F.6.1	Send E-mail	EcSbSubServer	Science User		The SBSRV will notify the Science User that an AST_EXP granule and associated signal file is available. The UR of the granule is passed in the notification to the user, along with a reference to the subscription that is being fulfilled.

This page intentionally left blank.

Abbreviations and Acronyms

ACL	access control list
ACMHW	access control and management HWCI
ADC	affiliated data center
ADSHW	advertising service HWCI
ADSRV	advertising service CSCI
AI	artificial intelligence
AI&T	algorithm integration and test
AIT	Algorithm Integration Team
AITHW	algorithm integration and test HWCI
AITT	Algorithm Integration and Test Team
AITTL	algorithm integration and test tools (CSCI)
AM-1	EOS AM Project spacecraft 1, morning spacecraft series -- ASTER, CERES, MISR, MODIS and MOPITT
ANSI	American National Standards Institute
API	application program (or programming) interface
AQAHW	algorithm QA HWCI
AQAHWCI	algorithm quality assurance hardware configuration item
ASCII	American Standard Code for Information
ASF	University of Alaska Synthetic Aperture Radar (SAR) Facility
AST	algorithm support team
ASTER	Advanced Spaceborne Thermal Emission and Reflection Radiometer
AVHRR	Advanced Very High-Resolution Radiometer
BAAS	Billing and Accounting Application Service
BOA	basic object adapter
BPS/bps	bits per second
Bps/bps	bytes per second

CASE	computer aided software engineering
CCS	CSMS communications system (CSCI)
CD	compact disk
CD-ROM	compact disk - read only memory
CDHF	central data handling facility
CDR	Critical Design Review
CDRL	Contract Data Requirements List
CERES	Clouds and Earth's Radiant Energy System
CHUI	Character User Interface
CI	configuration item
CIDM	client, interoprability and data management
CLS	Client Subsystem
CORBA	common object request broker architecture
COSS	common object services specifications
COTS	commercial off-the-shelf (hardware or software)
CPU	central processing unit
CS	computer software
CS	client server
CSC	computer software component
CSCI	computer software configuration item
CSDTs	computer science data types
CSMS	Communications and Systems Management Segment (ECS)
CSS	communication subsystem (CSMS)
DAA	DAN acknowledge
DAA	data availability acknowledgment
DAAC	distributed active archive center
DADS	data archive and distribution service
DADS	data archive and distribution subsystem (Version 0 system)
DADS	data archive and distribution system

DAN	data availability notice
DAO	data assimilation office
DAS	data assimilation system (at DAO)
DAS	data availability schedule
DBA	database administrator
DBMS	database management system
DCCI	distributed computing software configuration item
DCE	distributed communication environment
DCE	distributed computing environment (OSF)
DCF	data capture facility
DCHCI	distributed communications hardware configuration item
DCHCI	distributed computing hardware CI
DCHCI	distributed computing hardware configuration item
DDA	data delivery acknowledgment
DDICT	data dictionary CSCI
DDIST	data distribution services CSCI
DDL	data definition language
DDN	data delivery notice
DDSRV	SDPS - Document Data Server CSCI in the Data Server Subsystem
DEF	data exchange format
DEM	digital elevation model
DES	data encryption standard
DESKT	Desktop CI
DESKT	Desktop CSCI
DFS	distributed file system
DIB	directory information base
DID	data item description
DIM	distributed information manager
DIMGR	distributed information manager

DIMGR	distributed information manager CSCI
DIPHW	distribution and ingest peripheral HWCI
DIPHW	distribution and ingest peripheral management HWCI
DIS	data information system
DLPDU	data link protocol data unit
DM	data management
DMGHW	data management HWCI
DMS	data management subsystem
DNS	domain name system
DNS	domain name services
DOC	distributed object computing
DOF	distributed object framework
DPR	data processing request
DPREP	data pre-processing CSCI
DPRHW	data repository HWCI
DPS	data processing subsystem
DS	data server
DSS	data server subsystem
e-mail	electronic mail
email	electronic mail
EAS	ECS Advertising Service
Ecom	EOS Communications
Ecom	EOSDIS Communications System
ECS	EOSDIS Core System
EDC	EROS Data Center (DAAC)
EDF	ECS development facility
EDOS	EOS Data and Operations System
EDOS	EOSDIS Data and Operations System
EDU	EDOS Data Unit

EMC	enterprise monitoring and coordination
EOS AM	EOS AM Project (morning spacecraft series)
EOS	Earth Observing System
EOS-AM-1	EOS Morning Crossing (Descending) Mission
EOS-PM	EOS Afternoon Crossing (Ascending) Mission (afternoon spacecraft series) (see AIRS, AMSU-A, MHS, MIMR, CERES and MODIS)
EOSDIS	Earth Observing System (EOS) Data and Information System (DIS)
EOSDIS	Earth Observing System Data and Information System
ERD	entity relationship diagram
EROS	Earth Resources Observation System
ESA	European Space Agency
ESDD	Earth Science Data Directory
ESDIS	Earth Science Data and Information System (GSFC Code 505)
ESDT	Earth science data types
ESFU	enhanced standard format unit
ESH	EDOS service header
ESH	EDU service header
ESN	EOSDIS Science Network (ECS)
ETM+	Enhanced Thematic Mapper Plus
FDDI	fiber distributed data interface
FDF	flight dynamics facility
FOS	Flight Operations Segment (ECS)
FOT	Flight Operations Team
FSMS	File Storage Management System
FTP	file transfer protocol
ftpd	file transfer protocol daemon
G/B	gateway/bridge
GAC	global area coverage (AVHRR)
Gb	gigabits (109)

Gbps/GBps	gigabit/gigabyte per second
GByte	gigabyte (109)
GCDIS	Global Change Data and Information System
GCMD	Global Change Master Directory
GCP	ground control point
GDAO	GSFC Data Assimilation Office
GFE	Government furnished equipment
GIS	geographic information system
GNMP	GOSIP network management protocol
GOES	Geostationary Operational Environmental Satellite
GSFC	Goddard Space Flight Center
GTWAY	Version 0 Interoperability Gateway CSCI
GUI	graphical user interface
GV	ground validation
GV	TRMM Ground Validation Data
GV	TRMM Ground Verification
H&S	Health and Safety
H/K	housekeeping
H/W	hardware
HCL	Hughes class library
HDF	hierarchical data format
HP	Hewlett-Packard Corporation
HTML	hypertext markup language
HTTP	hypertext transport protocol
HWCI	hardware configuration item
I/F	interface
I/O	input/output
ICD	interface control document
ICL	ingest client

ICLHW	Ingest Client HWCI
ICMP	Internet control management protocol
ICMP	Internet control message protocol
ICMP	Internet control messaging protocol
IDD	interface definition document
IDL	interactive data language
IDL	interface definition language
IGTWY	External Communications Gateway
IMS	information management service
IMS	information management subsystem
INCI	internetworking CI
INGST	ingest services CI
INHCI	internetworking hardware CI
INS	SDPS Ingest Subsystem
IOS	interoperability subsystem
IOT	Instrument Operations Team
IP	Internet protocol
IR-1	interim release-1
IRD	interface requirements document
IRS	interface requirements specification
IS	ingest subsystem
ISS	internetworking subsystem of CSMS
JPL	Jet Propulsion Laboratory (DAAC)
KB	kilobyte (103)
KB	kilobytes
Kb	kilobit (103)
KB/SEC	kilobyte per second
Kbps	kilobits per second
Kbps/KBps	kilobit/kilobyte per second

Kerberos	security protocol developed by MIT; base for DCE security
Kftp	Kerberized file transfer protocol
KM	key mechanism
KSLOC	thousand single lines of code
Ktelnet	Kerberized telnet
L-7	Landsat-7 (Landsat-7 for EDHS search)
L0	Level 0
L0-L4	Level Zero through Level 4
L0R	Level 0 Reformatted
L0–L4	Level 0 through Level 4 data
LAC	Local Area Coverage (AVHRR)
LAN	local area network
Landsat	Land Remote-Sensing Satellite
LaRC	Langley Research Center (DAAC)
LIMGT	Local Information Manager CSCI
LIS	Lightning Imaging Sensor
M&O	maintenance and operations
MACI	management agent software CI
MACI	management agents configuration item
MB	megabyte (106 bytes)
Mb	megabit (106)
MBps	megabytes per second
Mbps	mega bits per second
MBPS/Mbps	million bits per second
Mbps/MBps	megabit/megabyte per second
Mbyte	megabyte
MCF	metadata configuration file
MCI	Management Software CI
med	medium

MEM	memory management
MET	metadata
Metadata	data about data
MFLOP	Million floating point operations per second
MFLOPS	mega (millions of) floating-point operations (106) per second
MHCI	management hardware CI
MHCI	management hardware configuration item
MIB	management information base
MIL-STD	military standard
min	minute
MIPS	mega (millions of) instructions (106) per second
MMI	man-machine interface
MO&DSD	Mission Operations and Data Systems Directorate (GSFC Code 500)
MOU	memorandum of understanding
MSFC	Marshall Space Flight Center (DAAC)
MSS	management subsystem service -- 214-001
MSS	Multispectral Scanner (Landsat)
MSS	system management subsystem (of CSMS)
MSSHW	MSS Hardware CI
MSU	mass storage unit
MSU	Microwave Sounding Unit
MTA	message transfer agent
MTTR	mean time to repair
MTTR	mean time to restore
MUI	management user interface
Multicast	a point to multi-point data flow
NASA	National Aeronautics and Space Administration
Nascom	NASA Communications
NCAR	National Center for Atmospheric Research

NESDIS	National Environmental Satellite, Data, and Information Service (NOAA)
NetCDF	non-HDF formats
netCDF	network common data format
NFS	network file system
NMC	National Meteorological Center (NOAA)
NMS	network management subsystem (Ecom)
NNTP	Network New Transfer Protocol
NNTP	Network News Transfer Protocol
NOAA	National Oceanic and Atmospheric Administration
NODC	National Oceanographic Data Center [NOAA] (also NESDIS/NODC)
NOLAN	Nascom Operational Local Area Network
NQS	queuing system
NRC	National Research Council
NRDN	NOAA Radar Data Network
NREN	National Research and Education Network
NRL	Naval Research Laboratory
NSF	National Science Foundation
NSFNet	NSF Network
NSI	NASA Science Internet
NSIDC	National Snow and Ice Data Center (DAAC)
NSIDC	National Snow and Ice Data Center (University of Colorado)
NSSDC	National Space Science Data Center
NTP	network transport protocol
NWCI	networking configuration item
O&M	operations and maintenance
O/A	orbit/altitude
ODBMS	object oriented data management system
ODC	other data center
ODL	object description language

ODMS	object data management system
OES	object encapsulation software
OMA	object management architecture
OMF	object management framework
OMG	object management group
OMT	Object Modeling Technique
OO	object oriented
OOA	object oriented analysis
OOD	object oriented design 12/94
OODBMS	object oriented database management system
OODCE	object oriented DCE
OODCE	object oriented distributed computing environment
OORDB	object oriented relational data base
OPS	operations
ORB	object request broker
ORDBMS	object rational database management system
OS	object services
OS	operating system 12/94
OSF	Open Software Foundation
OSI	Open System Interconnect
OSI	Open Systems Interconnection
OSI-RM	OSI reference model
OTS	off-the-shelf
P/L	payload
P/S	planning/scheduling
PACOR	packet processor
PAS	planning and scheduling
PCL	planning class libraries
PDF	publisher's display format

pdf	portable document format
PDL	program design language
PDPS	planning and data processing system
PDPS	planning and data production system
PDPS	product development and processing system
PDR	Preliminary Design Review
PDRA	Preliminary Design Review Release-A
PDS	planetary data system
PDS	platform data system
PDS	production data set
Perl	a UNIX® programming language
PGE	product generation executive (formerly product generation executable)
PGS	Product Generation Service
PGS	Product Generation System (ECS) (ASTER)
PGS	Product Generation System (obsolete ECS element name) (ASTER)
PGSTK	Product Generation System Toolkit
PI	Primary Investigator
PI	principal investigator
PI/TL	principal investigator/team leader
PLANG	production planning CSCI
PLNHW	planning HWCI
PLS	planning subsystem
POSIX	Portable Operating System Interface for Computer Environments
PR	Precipitation Radar (TRMM)
PRONG	Processing CSCI
PSCN	Program Support and Communications Network
PVL	parameter value language
Q/A	quality assurance
Q/A	quality/accounting

Q/L	quick look
R/T	real time
R/W	Read/Write
RAID	redundant array of inexpensive disks
RAM	random access memory
RCL	resource class libraries
RCP	remote copy
RDA	remote database access
RDBMS	relation data base management system
RDBMS	relational database management system
RID	review item discrepancy
RMA	reliability, maintainability, availability
RMON	remote monitor (SNMP)
RPC	remote procedure call
RPC	remote processing computer
RT	real time
RTM	requirements traceability model
S/C	spacecraft
S/E	systems engineering
S/W	software
SAA	Satellite Active Archives (NOAA)
SAA	satellite active archive
SBA	Small Business Administration
SBUV	Solar Backscatter Ultraviolet
SBUV/2	Solar Backscatter Ultraviolet/version 2
SCDO	Science and Communications Development Office (ECS)
SCF	Science Computing Facility
SDP	science data processing
SDPF	Sensor Data Processing Facility (GSFC)

SDPS	Science Data Processing Segment (ECS)
SDPS/W	science data processing software
SDPS/W	science data production software
SDPTK	science data processing toolkit
SDPTK	SDP toolkit
SDR	System Design Review
SDSRV	Science Data Server CSCI
SeaWiFS	Sea-Viewing Wide Field-of-View Sensor
SGI	Silicon Graphics Incorporated
SMC	system monitoring and coordination center
SMMR	Scanning Multichannel Microwave Radiometer
SMTP	simple mail transfer protocol
SNDCP	subnetwork dependent convergence protocol
SNICP	subnetwork independent convergence protocol
SNMP	simple network management protocol
SP	scenario primitive
SPRHW	science processing hardware CI
SSI&T	science software integration and test
SSM/I	Special Sensor Microwave/Imager
SSM/T	Special Sensor Microwave/Temperature Sounder
STMGT	storage management software CSCI
StP	Software Through Pictures
StP/OMT	Software Through Pictures/Object Modeling Technique
SUN	Sun MicroSystems
SW	science workstation
SW	software
SWCI	software configuration item
SWG	science working group
TBD	TO BE DEFINED

TBD	to be determined
TBR	to be resolved
TBS	to be supplied
TCP	transmission control protocol
TCP/IP	transmission control protocol/internet protocol
TDRSS	Tracking and Data Relay Satellite System
TIROS	Television and Infrared Observation Satellite
TL	team leader
TLCF	team leader computing facility
TM	Thematic Mapper (Landsat)
TMI	TRMM Microwave Imager
TN	TDRSS network
TOMS	Total Ozone Mapping Spectrometer
TOPEX	Ocean Topography Experiment (joint US-France)
TOVS	Television Infrared Observing Satellite (TIROS) Operational Vertical Sounder
TOVS	TIROS Operational Vertical Sounder
TRMM	Tropical Rainfall Measuring Mission (joint US-Japan)
TSDIS	TRMM Science Data and Information System
TSU	TRMM Science User
UARS	Upper Atmosphere Research Satellite
UDP	user datagram protocol
UDP/IP	user datagram protocol/internet protocol
UFS	UNIX file system
UID	universal identifier
UNIX	POSIX operating system
UR	universal reference
URL	universal reference location
URL	universal research locator
URL	universal resource locator

UserDIS	User Data Information System
USGS	U.S. Geological Survey
UT	universal time
UTC	universal time code
UTC	universal time coordinated
UTCF	universal time correlation factor
UTM	universal transverse mercator
UUID	universal unique identifier
UX	UNIX/X
V&V	verification and validation
V0 ODL	Version 0 Object Description Language
V0	Version 0
VAS	VISSR Atmospheric Sounder (GOES)
VIMS	virtual IMS
VIRR	Visible and Infrared Radiometer
VIRS	Visible Infrared Scanner (TRMM)
VIS	vendor information system
VIS-UV	Visible/Ultraviolet Spectrometer
VISSR	Visible/Infrared Spin-Scan Radiometer (GOES)
VT	virtual terminal
W/S	workstation
WAIS	Wide Area Information Server
WAN	wide area network
WKBCH	Workbench
WKBCH	Workbench CI
WKBCH	Workbench CSCI
WKSHC	Working Storage HWCI
WKSHW	Working Storage HWCI
WRKSTN	workstation

WS	working storage
WS	workstation
WWW	World Wide Web
X	X protocol
XTE	X-Ray Timing Explorer

This page intentionally left blank.